

iNEWS™

*Avstar Newsroom Computer System*

*Operations Manual*

*Version 1.3*



## Copyright and Disclaimer

Copyright © 2000 iNEWS™ All rights reserved. Printed in the U.S.A. All iNews products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specifications and price change privileges reserved.

The software described in this document is furnished under a license agreement and is protected under the copyright laws of the United States and other countries.

U.S. GOVERNMENT USERS RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restriction as set forth in subparagraph (b)(2) of the Technical Data and Computer Software-Commercial items clause at DFARS 252.211-7015, or in subparagraph (c)(2) of the Commercial Computer Software-Restricted Rights clause at FAR 52.227-19, as applicable.

Microsoft, the Microsoft logo, MS, MS-DOS, Win 32, Windows, Windows NT, Windows 2000, Windows NT Server, and the Windows operating system logo are registered trademarks of Microsoft Corporation in the United States of America and other countries. All other trademarks and registered trademarks used herein are the property of their respective owners.

iNEWS  
6400 Enterprise Lane  
Madison, Wisconsin 53719 USA  
Tel: +1-608-274-8686 Fax: +1-608-273-5876

iNEWS  
Intec 1  
Wade Road  
Basingstoke Hants RG24 8NE UK  
Tel: +44 1256 814300 Fax: +44 1256 814700

iNEWS  
Unit 6  
2 Eden Park Drive  
North Ryde NSW 2113 AUSTRALIA  
Tel: +61 2 8877 6880 Fax: +61 2 8877 6881

iNEWS  
Tegel Forum  
Breitenbachstraße 10  
Berlin 13509 GERMANY  
Tel: +49 30 5900993 0 Fax: +49 30 5900993 24

Avstar NRCS Operations Manual Version 1.3  
Document # 0130-00869 Rev. C  
October 3, 2000

# Contents

## Preface

Who Should Use This Manual . . . . .	xxvii
About This Manual . . . . .	xxviii
Sections . . . . .	xxviii
Symbols and Conventions . . . . .	xxviii
Structure of Text . . . . .	xxviii
Cross References . . . . .	xxix
Keyboard Conventions . . . . .	xxix
Console Conventions . . . . .	xxx
If You Need Help . . . . .	xxxi
...In Performing a System Operation . . . . .	xxxi
...With the Syntax of Console Commands . . . . .	xxxi
...With UNIX, or Specific Devices . . . . .	xxxii
Other Documentation . . . . .	xxxiii
Avstar Newsroom Computer System Documentation . . . . .	xxxiii
Broadcast Control System Documentation . . . . .	xxxiii
Other Products . . . . .	xxxiv
If You Have Documentation Comments . . . . .	xxxiv

## Section I

## Avstar Overview and System Basics

### Chapter 1

### Introduction

What is Avstar? . . . . .	1-2
iNEWS Products . . . . .	1-3
Avstar Newsroom Computer System . . . . .	1-4
iNEWS Media Browse 2000 . . . . .	1-5
iNEWS Broadcast Control System . . . . .	1-6
Links to Other Newsroom Products . . . . .	1-6

---

System Administrator Tasks . . . . .	1-8
Basic System Administration Tasks . . . . .	1-8
User Tasks . . . . .	1-8
Database Tasks . . . . .	1-9
Security Tasks . . . . .	1-9
Customizing Commands and Messages . . . . .	1-10
Storage Maintenance Tasks . . . . .	1-10
Device Tasks . . . . .	1-10
Reviewing Default Settings . . . . .	1-11
Troubleshooting . . . . .	1-11

## Chapter 2

### The Avstar Console

Overview . . . . .	2-2
Commands You Can Type at the Console . . . . .	2-3
Console Control Commands . . . . .	2-3
Example: The Computer Command . . . . .	2-4
Selecting Servers . . . . .	2-5
Selecting One or More Servers . . . . .	2-5
Zooming in on One Server . . . . .	2-6
Console History . . . . .	2-7
Pausing the Screen Display . . . . .	2-8
Viewing Recent Console History . . . . .	2-8
Reading Older History . . . . .	2-10
Console Function Keys . . . . .	2-11
Assigning a Command to a Function Key . . . . .	2-11
Changing the Assignment of a Function Key . . . . .	2-11
Deleting the Definition of a Function Key . . . . .	2-11
Displaying Function Key Assignments . . . . .	2-12
Console Operations . . . . .	2-12
If the Console Freezes . . . . .	2-12
Exiting the Console . . . . .	2-13
Starting the Console . . . . .	2-14
The Remote Console . . . . .	2-14

Dialing in to the Console .....	2-14
Executing Commands Remotely.....	2-15
Logging out from a Remote Console .....	2-16
Logging out a Remote User from the Main Console .....	2-16
The Console Configuration File (console.cfg) .....	2-16
Looking at the Console Configuration File .....	2-17
Editing the Configuration File.....	2-18
Console Configuration Keywords .....	2-19
Console Control Command Reference.....	2-21

## Chapter 3

### Getting Started

Logging in as System Operator .....	3-2
Becoming a Console Superuser .....	3-2
Entering Superuser Mode .....	3-3
Exiting Console Superuser Mode .....	3-3
Changing the System Administration Passwords .....	3-3
Selecting Passwords .....	3-3
Changing the System Operator Password.....	3-4
Changing the Superuser Password.....	3-4
Startup and Shutdown .....	3-5
Starting the System .....	3-5
Shutting Down the System .....	3-9

## Chapter 4

### Users

Viewing User Accounts .....	4-2
Modifying User Traits.....	4-3
Modify User Account Dialog Box .....	4-4
User Traits Summary .....	4-5
User Name.....	4-5
Type .....	4-5
Edit Mode .....	4-6
Queues.....	4-6
Get from Template.....	4-9

---

Changing a User's Password . . . . .	4-9
Changing User Preferences . . . . .	4-10
Preferences Dialog Box . . . . .	4-11
Session Tab. . . . .	4-11
Confirmations Tab. . . . .	4-12
Backup Tab. . . . .	4-14
Refresh Tab. . . . .	4-15
Layout Tab . . . . .	4-16
Search Results Tab. . . . .	4-19
Simplified User Settings. . . . .	4-19
Simplified User Setting Dialog Box. . . . .	4-20
Workspaces . . . . .	4-21
Application . . . . .	4-21
Setting up New Users in Avstar . . . . .	4-22
Creating a New User Area in the News Database . . . . .	4-23
Adding a New User Account . . . . .	4-26
Copying User Traits to Another User Account . . . . .	4-26
Creating a New User Account . . . . .	4-27
Enabling a New User to Receive Mail . . . . .	4-29
Searching for Information About Users . . . . .	4-29
Removing User Accounts . . . . .	4-33
Creating a User Manager Account . . . . .	4-35
Creating a Database Manager Account . . . . .	4-36
Logging out All Users . . . . .	4-37

## Chapter 5

### Stories, Queues, and Directories

Overview . . . . .	5-2
Adding a Directory or Queue . . . . .	5-3
A Few Restrictions:. . . . .	5-3
Creating a New Directory. . . . .	5-4
Creating a New Queue . . . . .	5-6
Setting up the Outgoing Mail Queue . . . . .	5-7
Setting up the Dead Letter Queue . . . . .	5-8

Creating a New Story.....	5-9
Removing a Directory or Queue .....	5-9
Renaming a Directory or Queue.....	5-10
Viewing Database Traits.....	5-13
From the Avstar Workstation....	5-13
From the Avstar Console....	5-14
Sending Output from the List Command to a Printer. . . .	5-15
Getting Information about Stories .....	5-15
Finding out Who Moved, Duplicated, or Killed a Story ..	5-17
Recovering a Killed Story.....	5-20
Changing Database Traits .....	5-21
Database Traits Summary .....	5-25
Directory/Queue Properties Dialog Box .....	5-25
Forms Tab.....	5-27
Index Field/Story Form Compatibility Error Messages ..	5-30
Starting the Queue Sort Function .....	5-33
Groups Tab.....	5-34
Abstract Tab.....	5-36
Uses for Abstract Printing.....	5-37
Maintain Tab .....	5-38
Choosing Queues to be Purged .....	5-40
Choosing a Purge Interval .....	5-41
Matching Purge Intervals .....	5-41
Purge Intervals and the Purge Limit.....	5-42
User Interface Tab.....	5-43
Locks Tab .....	5-46
Locking and Unlocking.....	5-47
Identifying Locked Queues and Stories .....	5-47
From the Avstar Workstation....	5-47
From the Avstar Console....	5-48
Finding out Who Last Locked a Queue .....	5-48
Finding out Who Last Ordered a Queue .....	5-48

Finding out Who Last Locked the Story .....	5-49
Types of Locks .....	5-50
Edit Lock .....	5-50
Removing a Story's Edit Lock .....	5-50
User Lock .....	5-51
Removing a Story's User Lock Without a Key .....	5-51
Removing a Queue's User Lock Without a Key .....	5-52
Order Lock .....	5-52
Production Lock .....	5-53
Unbusy Stories and Queues .....	5-53

## Chapter 6

### Groups

Overview .....	6-2
Viewing Group Information .....	6-2
From the Console... .....	6-2
From a Workstation... .....	6-3
Creating a New Group .....	6-5
Step 1 - Choosing a Group Name .....	6-5
Step 2- Enter Group Name in System .....	6-6
Step 3- Specifying Members of New Group .....	6-6
The Group Checker .....	6-7
Group Checker Error Messages .....	6-9
Renaming a Group .....	6-12
Step 1- Change Group Name in System .....	6-12
Step 2- Change Group Name in SYSTEM.GROUPS .....	6-12
Deleting a Group .....	6-13
Creating or Modifying Multiple Groups in Interactive Mode .....	6-14
Adding Members to an Existing Group .....	6-15
Users as Members of a Group .....	6-15
Groups as Members of Other Groups .....	6-15
Avoiding Recursion .....	6-16
Workstations as Members of Groups .....	6-17



Combining Permissions .....	6-18
Being More Restrictive .....	6-18
Being Less Restrictive .....	6-19
Setting an Automatic Timeout .....	6-19
Modifying Idle Timeout .....	6-19
Modifying Login Timeout .....	6-20
Timeout Value Settings and Format .....	6-20
Group Access and Usage Restrictions .....	6-20
Access and Usage Examples .....	6-21
Group Traits for the Database .....	6-22
Read Group .....	6-23
Write Group .....	6-24
Notification Group .....	6-25
Restricting Both Reading and Writing .....	6-26
Transferring Group Assignments .....	6-27
Hiding Queues and Directories .....	6-27
Mail Aliases .....	6-28
Creating a Mail Alias .....	6-29
Mail Aliases for Other Machines or the Internet .....	6-30

## Chapter 7

### Keyboards and Macros

Understanding Macros and Keyboards .....	7-2
Customizing Workstation Keyboards .....	7-3
Creating a New Keyboard Story .....	7-3
Creating a Macro .....	7-5
Adding Comments .....	7-6
Assigning Macros to Keys .....	7-6
Predefined System Function Keys .....	7-8
The State Keys .....	7-9
Using Plain Text in Macros .....	7-11
Repeating Macros .....	7-11
Notes of Caution for Creating Macros .....	7-12
Keyboard Checker .....	7-13

Error Messages .....	7-13
Warning Messages .....	7-14
Assigning a Default Keyboard to a User .....	7-16
Customizing Keyboards for VT/DOS Terminals .....	7-19
Miscellaneous VT Macro Tips .....	7-20
The Pause Command.....	7-20
The Blank Command.....	7-20
Repeating VT Macros.....	7-20
Assigning VT Macros to Standard Macro Keys .....	7-22
Extended Versus Standard Macro Keys .....	7-23
Enabling F13 on the VT220 .....	7-24
Assigning VT Macros to Extended Macro Keys.....	7-24
For VT Users Who Switch Between Keyboards .....	7-26

## Chapter 8

### Forms

Form Names and Locations.....	8-2
Guidelines for Designing Forms.....	8-2
Creating a Form.....	8-2
Customizing Forms .....	8-6
Label Borders .....	8-11
Assigning a Form as a Queue or Story Form .....	8-11
Form Field Types and Definitions .....	8-13
Avstar MCS/BCS Fields and Forms .....	8-24
Standard Avstar Forms .....	8-27
Account Queue Form .....	8-27
Mail Form .....	8-29
Timing Form.....	8-29
Print Form.....	8-30
Seek Form .....	8-31
Wire Story Form.....	8-33
Mapping Netstation Characters to Avstar .....	8-34

<b>Section II</b>	<b>System Setup and Configuration</b>	
<b>Chapter 9</b>	<b>Character Generator</b>	
	<b>Title Entry</b>	
	Overview of CG Title Entry . . . . .	9-2
	Title Entry Setup and Configuration . . . . .	9-4
	Understanding CG Templates . . . . .	9-4
	CG Template Backgrounds . . . . .	9-6
	Required Bitmaps . . . . .	9-6
	Capture Tool . . . . .	9-7
	Hardware Requirements for Capture Tool . . . . .	9-7
	Installation of Capture Tool . . . . .	9-8
	Using the Capture Tool . . . . .	9-8
	CG Template Editor . . . . .	9-11
	Edit Title Entry Template Window . . . . .	9-11
	Creating a New Template . . . . .	9-15
	Using Font PreSets . . . . .	9-20
	Title Entry Security . . . . .	9-23
	Access to CG Template Editor . . . . .	9-23
	Access to CG Title Entry . . . . .	9-23
<b>Chapter 10</b>	<b>ed, the Line Editor</b>	
	Overview - Before Editing . . . . .	10-2
	Making a Backup File Before Editing . . . . .	10-2
	Viewing the Contents of a File . . . . .	10-2
	Printing a Copy of a File . . . . .	10-4
	Using the UNIX Line Editor . . . . .	10-4
	Launching ed . . . . .	10-5
	Specifying Lines to Edit . . . . .	10-5
	Searching the File . . . . .	10-7
	Searching Tips . . . . .	10-8
	Editing Commands . . . . .	10-9
	Saving Changes . . . . .	10-14

---

Quitting ed .....	10-14
-------------------	-------

## Chapter 11

### Configuration Files

Licensing of Avstar System Components .....	11-2
Device Types .....	11-3
Viewing Information About Your Devices .....	11-4
List C Message Columns .....	11-5
The Site Configuration File (/site/config) .....	11-7
Testing the Site Configuration File After Changing .....	11-9
Incorporating Configuration Changes .....	11-10
Changing the Configuration File .....	11-15
The Hosts File (/etc/hosts) .....	11-17
System Profile Files .....	11-17
Viewing the System Profile File .....	11-18
Changing the System Profile File .....	11-19
Listing Parameter Settings .....	11-20
System Profile Parameters .....	11-21
Adding Devices to Your Avstar System .....	11-29
Adding a PCU .....	11-30
PCU Device Numbering .....	11-34
Adding a Workstation .....	11-35
Adding a DOS PC Workstation .....	11-39
Phase 1 .....	11-39
Phase 2 .....	11-39
Phase 3 .....	11-40
Adding a Printer .....	11-41
Adding a Wire .....	11-47
Alternative Editing of the Site Configuration File .....	11-48
Intersystem Messaging .....	11-51
Sending Intersystem Messages .....	11-51
Receiving Intersystem Messages .....	11-53
Database Change .....	11-56
Avstar Workstation Session Behavior .....	11-56

VT Session Behavior .....	11-56
---------------------------	-------

## Chapter 12

### Printers

System Printing .....	12-2
The Printer Profile Files (in /site/printers) .....	12-2
Customizing Print Effects (Fonts) .....	12-4
Defining a Font .....	12-4
Combining Print Effects .....	12-4
Defining Print Forms .....	12-5
Defining a Form .....	12-6
Combining Setup Options .....	12-6
Font and Form Space Available .....	12-7
Printer Profile Options .....	12-7
Profile-Only Options .....	12-8
Profile and Style Options .....	12-13
Standard Header and Footer Options .....	12-15
User-Selected Headers and Footers .....	12-16
Profile Option Defaults .....	12-17
Using Special Characters in a Profile .....	12-19
Using Nonprinting Characters .....	12-19
Adding Nonprinting Characters by Alias .....	12-19
Adding Nonprinting Characters by Decimal Value .....	12-20
Avoiding Characters Used by the System .....	12-20
Creating and Using Print Styles .....	12-23
Creating a Style Story .....	12-24
Changing System Profile Options .....	12-25
Selecting Forms .....	12-26
Identifying and Selecting Fonts .....	12-26
Using Styles with Local Printing on Video Terminal Only .....	12-30
Local Printing .....	12-31
Local Printing Dialog Box .....	12-32
Scope .....	12-33

Story Preview . . . . .	12-34
Options . . . . .	12-35
Copies . . . . .	12-36
Grid . . . . .	12-37
Print Preview and Network buttons . . . . .	12-37
Local Print Style Options . . . . .	12-38
Banner Format Options . . . . .	12-44
Example Style Story . . . . .	12-46
Managing Printers . . . . .	12-47
Removing a Pending Print Request . . . . .	12-47
Restarting the Current Print Request . . . . .	12-48
Reordering a Pending Print Request . . . . .	12-48
Cancelling a Runaway Print Job . . . . .	12-48
Responding to a “Printer Offline” Problem . . . . .	12-48

## Chapter 13

### Wires

Adding a Wire to Your Avstar System . . . . .	13-2
Phase 1 . . . . .	13-2
Phase 2 . . . . .	13-2
Phase 3 . . . . .	13-7
Phase 4 . . . . .	13-9
Wires Profile Files . . . . .	13-11
Wire Profile Options . . . . .	13-13
Using Special Characters in a Profile . . . . .	13-26
Entering Nonprinting Characters . . . . .	13-26
Entering Characters by Alias . . . . .	13-26
Entering Characters by Decimal Value . . . . .	13-27
Avoiding Characters Used by the System . . . . .	13-27
Converting Text with Accents and Diacritical Marks . . . . .	13-28
Distributing Stories from the Wire . . . . .	13-31
Defining Distribution of Wire Stories . . . . .	13-31
Creating a Distribution Name . . . . .	13-32
Identifying a Destination Queue . . . . .	13-33

Changing Notification Priority .....	13-33
Setting the Transmit or Always Options .....	13-34
Adding a Distribution Line .....	13-35
Avoiding Hidden Categories .....	13-35
Using the WIRES.ALL Notification Priority .....	13-36
Distributing Unknown Wires .....	13-37
Maximum Number of Lines .....	13-38
Mailboxes .....	13-38
Purge Intervals .....	13-38
Internationalization .....	13-39
Operating Wire Keyword Searches .....	13-39
Setting up Keyword Searching .....	13-39
Additional Information about Search Jobs .....	13-42
Suppressing a Search .....	13-42
Default Entry .....	13-42
Keyword Limitations .....	13-43
Keyword Searching .....	13-43
Using Parentheses in Searches .....	13-45
Tips on Building Search Rules .....	13-46
User Notification .....	13-47
Removing a Rule Set .....	13-47
Sending a Story to More Than One Queue .....	13-48
Default Directory Paths .....	13-48
The Keyword Mailbox .....	13-48
Keyword Checker Messages .....	13-49

## Chapter 14

### Servers

Overview .....	14-2
Adding a Server Program to the System .....	14-3
Job Lists: Queues, Stories, and Commands .....	14-7
Defining Tasks for Servers .....	14-7
Adding a Scan Line in a Job List Story .....	14-9
Defining a Priority Queue .....	14-9

---

Defining an Every Entry Queue .....	14-10
Defining a Server Command Set .....	14-10
Processing Deleted Stories .....	14-12
Ordered Queues and the Order Command .....	14-13
Defining Mailbox Tasks .....	14-13
Using Mailboxes .....	14-14
Reserved Mailboxes .....	14-16
Assigning a Mailbox to a Queue .....	14-17
Defining Timed-Interval Tasks .....	14-18
Example of Timed-Interval Tasks .....	14-20
Action Servers .....	14-21
Adding an Action Server .....	14-22
Assigning Field Validation .....	14-26
Background and Possible Uses of Validation .....	14-26
Using Validation with Action Servers or Tx Links .....	14-27
Using the Validation Feature .....	14-28
Validation Job List Commands .....	14-29
Distribution Servers .....	14-30
Assigning Distribution Codes .....	14-32
Using Wildcards and the Destination Queue .....	14-33
From the Command Line .....	14-34
Using an Action Server or Tx Link .....	14-34
Using Dup or Move Commands in Job Lists .....	14-34
Using Validate Commands in Job Lists .....	14-35
Instructions in the Wire Distribution Story .....	14-35
Matching and Case .....	14-36
Matching and Order .....	14-36
Adding a Distribution Server .....	14-38
Parallel Wire Servers .....	14-41
Database Components for Backup Wire Distribution	14-42
Adding a Parallel Wire Server .....	14-43
Keyword Servers .....	14-51
Database Components for Expanded Wire Keyword	



---

Searches .....	14-52
Adding a Keyword Server .....	14-54
System Servers .....	14-60
Seek Servers .....	14-60
Installing a Seek Server .....	14-61
Fast Text Search (FTS) Servers .....	14-63
FTS Workflow .....	14-63
FTS Indexing .....	14-64
Installing FTS Components .....	14-66
...On the Windows NT Server .....	14-67
Installing <code>ftsidx.exe</code> and <code>ftssch.exe</code> .....	14-67
Starting <code>ftsidx.exe</code> and <code>ftssch.exe</code> .....	14-68
Stopping <code>ftsidx.exe</code> and <code>ftssch.exe</code> .....	14-69
...On Avstar Servers (UNIX) .....	14-69
Setting up <code>ftsseek</code> and <code>ftsindex</code> Servers .....	14-72
Starting and Stopping <code>ftsindex</code> and <code>ftsseek</code> on the Avstar Server .....	14-74
Batch Indexing .....	14-75
Batch Indexing Directories (Folders) .....	14-76
Dynamic Indexing .....	14-76
Archival and Backup .....	14-76
Copying and Archiving the Index Base Files .....	14-76
Removing the Index Base and Reindexing (Optional) .....	14-77
Print Servers .....	14-78
Adding a Print Server .....	14-79
Mail Servers .....	14-80
Disabling Mail to All Users .....	14-81
Using Network Mail .....	14-81
Using 8-Bit Characters in Mail .....	14-82
Character Conversion Table for Underscore-Prefix Format ... 14-83	
Networking Two or More Servers Using Rx/Tx Links .....	14-88
Sending Story Forms .....	14-89

Understanding the READY Field . . . . .	14-90
Setting Automatic Update . . . . .	14-90
Updating Queue Considerations . . . . .	14-91
Changing Queue Order . . . . .	14-92
Adding Rx/Tx Links . . . . .	14-93
Adding Network Links . . . . .	14-94

## Chapter 15

### Web Publishing

Overview . . . . .	15-2
Setting Up Tx/net to Send HTML . . . . .	15-2
Default HTML Skeleton Story Form and Queue . . . . .	15-4
Creating an HTML Story Form . . . . .	15-5
Adding Story Entity References . . . . .	15-5
NSML to HTML Conversion . . . . .	15-10
Web Story Directives . . . . .	15-14
Using Optional Format Strings . . . . .	15-14
Time and Date Elements . . . . .	15-15
Time and Date Fields . . . . .	15-17
Time and Date Field Example . . . . .	15-17
A Sample HTML Story Form . . . . .	15-18
Characteristics of the Story . . . . .	15-18
Contents of the Form . . . . .	15-20
Resulting HTML Output . . . . .	15-22

## Chapter 16

### Web Access

Overview . . . . .	16-2
Starting the Web Server . . . . .	16-2
Web Access Login . . . . .	16-3
Web Access Story Templates . . . . .	16-3
Default Story Template . . . . .	16-6
Web Access Directory and Queue Templates . . . . .	16-7
Template Entities . . . . .	16-8
Default Directory Template . . . . .	16-18

	Default Queue Template .....	16-19
	Web Access Configuration.....	16-20
<b>Section III</b>	<b>System Operations and Troubleshooting</b>	
<b>Chapter 17</b>	<b>Connect Services</b>	
	Network Services.....	17-2
	Dialogs for Connect Services .....	17-2
	Building a Dialog .....	17-2
	Dialog Commands .....	17-4
	Dialog Examples .....	17-5
	Adding System Services .....	17-5
	Setting up the Service.....	17-6
	Console Connect Sessions.....	17-11
	Serial Connect Services .....	17-13
<b>Chapter 18</b>	<b>Database Security</b>	
	Establishing Security Procedures .....	18-2
	User Passwords .....	18-3
	Checking Password Status .....	18-3
	Forcing Individual Users to Change Their Passwords.....	18-6
	... At an Avstar Workstation .....	18-6
	Tracking User Activity .....	18-7
	Tracking User Login Activity and Date Created.....	18-8
	... At an Avstar Workstation .....	18-8
	... At the Avstar Console.....	18-9
	Listing Users Currently Logged in .....	18-9
	Recording Logins .....	18-10
	Using Group Security to Control System Access .....	18-12
<b>Chapter 19</b>	<b>Database Management</b>	
	Monitoring Free Space .....	19-2
	Understanding Database Storage Units .....	19-3
	Monitoring the Free List .....	19-3

Understanding How the System Copies Stories . . . . .	19-4
Tracking Database Space over Time . . . . .	19-5
Using the hogs Command to Obtain Information . . . . .	19-5
Using dbpurge and dbfree to Obtain Information . . . . .	19-6
Increasing Database Space for Immediate Use. . . . .	19-7
Maintaining the Database . . . . .	19-8
Checking the Database for Errors . . . . .	19-8
Cleaning the Database. . . . .	19-9
Cleaning Your Database Offline. . . . .	19-10

## Chapter 20

### Backing up Your System

Tape Operations . . . . .	20-2
Establishing Policies for Backup Procedures. . . . .	20-2
Backing up the Avstar Database. . . . .	20-3
The dbdump Command . . . . .	20-4
Backing up Entire Database . . . . .	20-5
Backing up Individual Queues . . . . .	20-7
Notes on Backing up the Database . . . . .	20-9
Restoring Data to the Avstar Database . . . . .	20-10
The dbrestore Command . . . . .	20-10
Restoring a First-Level Directory . . . . .	20-10
Listing Tape Contents and Backup Dates. . . . .	20-11
Listing Contents of a Tape . . . . .	20-12
Listing Items Dumped on a Particular Date . . . . .	20-13
Listing the Date of Each Backup . . . . .	20-14
Searching a Tape. . . . .	20-14
The searchtape Command . . . . .	20-14
Searching a Tape for Stories. . . . .	20-15
Searching a Tape by Word(s). . . . .	20-15
Searching a Tape by Word and Date Range . . . . .	20-16
Searching a Tape by Word and Day . . . . .	20-17
Searching a Tape by Word and Month . . . . .	20-17
Specifying a Maximum Number of Stories to Search . . . . .	20-17

	Checking for Free Space on a Database . . . . .	20-18
	Adding Blocks to the Free List . . . . .	20-18
	Notes on Restoring the Database . . . . .	20-19
	Disaster Recovery Planning . . . . .	20-20
	Disaster Recovery Dbdump . . . . .	20-20
	Create Minimal dbdump. . . . .	20-20
	Disaster Recovery Dbrestore. . . . .	20-21
	Backing up Software . . . . .	20-22
	The softdump Command . . . . .	20-23
	Using the softdump Command to Back up System Software . . . . .	20-23
	SCO Emergency Boot and Root Floppies . . . . .	20-24
	Backing up Site Files . . . . .	20-25
	The sitedump Command. . . . .	20-25
<b>Chapter 21</b>	<b>Disconnects</b>	
	Normal System Status . . . . .	21-2
	Detecting a Disconnect . . . . .	21-3
	Types of Disconnects. . . . .	21-5
	Causes of Disconnects. . . . .	21-6
	Disconnect Recovery . . . . .	21-7
	Overview . . . . .	21-7
	Procedures . . . . .	21-8
	Worksheet . . . . .	21-17
<b>Chapter 22</b>	<b>Troubleshooting</b>	
	Avstar Workstation Problems . . . . .	22-2
	A User Cannot Log in . . . . .	22-2
	A User Cannot Establish a Session. . . . .	22-3
	A User Cannot Access an Item . . . . .	22-5
	Group Permissions. . . . .	22-5
	Busy Stories. . . . .	22-6
	Wire Problems . . . . .	22-7
	System Printer Problems. . . . .	22-9

Locked Blocks .....	22-11
How to Check Process Status(ps Command) .....	22-11
Power Failure .....	22-13
Hard Drive Failure .....	22-14
Network Failure .....	22-14
netstat -i Command .....	22-15
Output Errors (Oerrs) .....	22-15
Input Errors (Ierrs) .....	22-16

**Section IV****System Reference****Appendix A****Command References**

Programs Invoked by Avstar .....	A-2
Commands Used by iNews personnel Only .....	A-3
UNIX Commands Used in Avstar .....	A-4
Console Server Commands .....	A-4
Job List Commands .....	A-42
Dialog Commands .....	A-48

**Appendix B****System Files**

/etc/bootptab .....	B-2
SCO-UNIX Specific .....	B-2
SGI-Irix Specific .....	B-3
DEC/MIPS Specific .....	B-4
/etc/hosts .....	B-4
/etc/networks .....	B-5
/site/config .....	B-6
/site/printers/ti830 .....	B-8
/site/system .....	B-8
/site/wires/anpa7 .....	B-9
console.cfg .....	B-10
SYSTEM.CLIENT.WINDOWS .....	B-11
SYSTEM.CONFIGURE.301-ACTION .....	B-12
SYSTEM.MAP .....	B-13

SYSTEM.RESOURCE .....	B-15
SYSTEM.WIRES.DISTRIBUTION .....	B-17
SYSTEM.WIRES.KEYWORDS .....	B-18
SYSTEM.WIRES.KEYWORDS-AP .....	B-19
SYSTEM.WIRES.KEYWORDS-AP2 .....	B-19

## Appendix C

### Standard Dictionaries

Using Dictionaries to Define Messages and Commands .....	C-2
Customizing Dictionaries .....	C-4
Changing Default Dictionary Values .....	C-4
Restoring Dictionary Defaults .....	C-8
Utility Messages Dictionary (/site/dict/messages) .....	C-9
CCU Messages Dictionary (/site/dict/ccumsgs) .....	C-15
Commands Dictionary (/site/dict/ccucmds) .....	C-19
Console Commands .....	C-20
Job List Commands .....	C-25
Video Attribute Dictionary (/site/dict/ccuvideo) .....	C-26
Queues Dictionary (/site/dict/queues) .....	C-27
Words Dictionary (/site/dict/words) .....	C-29
Connect Dictionary (/site/dict/doac) .....	C-33
Telex Dictionary (/site/dict/telex) .....	C-37
Dial Dictionary (/site/dict/dial) .....	C-39
Keyboard Macros Dictionary (/site/dict/keymacros) .....	C-40
Printer Messages Dictionary (/site/dict/printmsgs) .....	C-43
Case-Shifting Dictionary (/site/dict/shift) .....	C-44
VT Map Dictionary (/site/dict/vtmap) .....	C-46

## Appendix D

### PCU/CCU Reference

Overview .....	D-2
Network CCUs .....	D-3
Connecting Devices to a CCU .....	D-4
Physical Specifications for CCU II and CCU III .....	D-5
Environmental Requirements .....	D-5

---

CCU II.....	D-6
Resetting a CCU II .....	D-7
Locating Panel Lights .....	D-7
Connecting Devices to a CCU II.....	D-8
Locating Ports .....	D-8
Connecting a CCU to the System .....	D-8
CCU III.....	D-10
Resetting a CCU III .....	D-10
Locating Panel Lights.....	D-11
Understanding the LED Display .....	D-11
POST Error Messages.....	D-12
Connecting Devices to a CCU III.....	D-14
Locating Ports .....	D-14
Connecting a CCU to the System .....	D-15
CCU III Backplane .....	D-15
Setting the Host Baud Rate DIP Switch.....	D-16
CCU IV.....	D-16
Resetting a CCU IV .....	D-16
Locating Front Panel Lights .....	D-17
Understanding the LED Display .....	D-18
Connecting Devices to a CCU IV.....	D-19
Locating Ports.....	D-19
Connecting a CCU IV to the System.....	D-20
CCU IV Interior.....	D-20
PCU.....	D-21
Resetting a PCU .....	D-21
Locating Front Panel Lights .....	D-22
Understanding the LED Display .....	D-22
PCU LED Codes .....	D-22
Connecting Devices to a PCU .....	D-24
Locating Ports .....	D-24
Connecting a PCU to the System.....	D-24



<b>Appendix E</b>	<b>Character Mapping Code Tables for Wires</b>	
	ASCII (7-bit) Character Set . . . . .	E-2
	IBM Character Set . . . . .	E-8
	dbrestore Conversion Map. . . . .	E-15
	Sample Arabic Wire Profile . . . . .	E-16
<b>Appendix F</b>	<b>Environment Variables</b>	
	Registry Editor . . . . .	F-2
	Environment Variables . . . . .	F-3
	CCColor . . . . .	F-3
	DestinationOrder . . . . .	F-5
	MailLookup . . . . .	F-7
	MsgMailAlert . . . . .	F-8
	PIColor . . . . .	F-10
	RGB Hexadecimal Color Chart . . . . .	F-11
	ShowTimingBar . . . . .	F-12
	Scan Codes . . . . .	F-13
	SyncToServer. . . . .	F-16
	VT Compatibility . . . . .	F-18
	Delete_Notify . . . . .	F-19
<b>Appendix G</b>	<b>Managing Traits at Console</b>	
	Viewing User Traits from the Console . . . . .	G-2
	Modifying User Traits from the Console . . . . .	G-3
	Users' Passwords. . . . .	G-4
	Listing Users Who Do Not Have Passwords . . . . .	G-7
	User Traits Console Command Summary . . . . .	G-11
	Managing Database Traits from the Console . . . . .	G-15
	Getting Basic Information . . . . .	G-15
	Getting Detailed Information . . . . .	G-16
	Changing Database Traits from the Console. . . . .	G-17
	Changing a Parent Directory Only . . . . .	G-17
	Database Traits Console Command Summary . . . . .	G-18

Sortfield .....	G-25
Changing a Queue's Sort Field .....	G-25
Starting the Queue Sort Function from the Console .	G-26
Purge Interval .....	G-27
Abstract Printing. ....	G-28
Abstract Printer .....	G-29
Abstract Lines .....	G-29
Abstract Style .....	G-30
Mailbox .....	G-31
The dis Column. ....	G-31
Preview Lines. ....	G-32
Refresh .....	G-33
Managing Group Traits at the Console .....	G-34
Groups .....	G-34
Read Group .....	G-34
Write Group .....	G-35
Notify Group .....	G-35
Restricting Access Using Read and Write Limitations. ....	G-36
Removing Directory or Queue Restrictions. ....	G-37

**Glossary**

**Index**

**Reader's Comments**

# *Preface*

This Operations Manual provides information on how to manage the Avstar Newsroom Computer System, consisting of:

- A console
- One or more servers
- Various clients, such as Avstar Workstations and printers

## **Who Should Use This Manual**

This manual is written for system administrators who are managing the Avstar Newsroom Computer System (NRCS). It is strongly recommended that system administrators have prior experience in or classroom knowledge of UNIX system administration. Avstar system administrators need to:

- Manage user accounts, security and permissions
- Start up and shut down Avstar NRCS
- Perform file system maintenance, backup, and recovery
- Maintain disks
- Monitor processes
- Configure and monitor the network

## About This Manual

This manual provides information in the following format:

### Sections

- Section I, *Avstar Overview and System Basics*, provides an overview of Avstar NRCS: information about the Avstar console; users, groups, directories, queues, and stories; keyboard macros; and forms. It contains Chapters 1-8.
- Section II, *System Setup and Configuration*, provides information about UNIX, printers, wires, servers, web publishing, and web access. It contains Chapters 9-16.
- Section III, *System Operations and Troubleshooting*, discusses connect services, system security, database management, backing up your system, and troubleshooting. It contains Chapters 17-22.
- Section IV, *System Reference*, contains a command reference, sample system files, information about standard dictionaries, PCU/CCU references, and character mapping codes for wires. It contains Appendices A-G, the Glossary, Index, and the Reader's Comments form.

## Symbols and Conventions

This manual uses the following special symbols and conventions:

### Structure of Text

1. Numbered lists, when the order of the primary items is important.
  - a. Alphabetical lists, when the order of secondary items is important or in the case of optional procedures.
- Bulleted lists, when the order of primary items is unimportant.

- Indented dashed lists, when the order of secondary items is unimportant.

Look here in the margin for tips and environment-specific information.

In the margin you will find tips that help you perform tasks more easily and efficiently. You will also find information specific to a particular operating environment.



*A note provides important related information, reminders, recommendations, and strong suggestions.*



**A caution means that a specific action you take could harm your computer or cause you to lose data.**

## Cross References

Cross references are provided throughout this manual to give readers locations where additional—sometimes more detailed—information on a certain topic can be found. In some cases, the chapter name and number is provided. In most cases, a two-part page number is given along with the name of a section header. The first number in the page number is actually the chapter number.

For instance: See “Changing Database Traits” on page 5-21 for more information.

See “About This Manual” on page xxviii for more information on what chapters are in which sections of this manual.

In this example, information on how to change database traits can be found on page 21 in Chapter 5, which is in Section I. Chapters are numbered consecutively; page number restart at one in each chapter. Section numerals are not provided in cross references. So, a cross reference that shows page number 17-2, for instance, indicates that the information is in Chapter 17.

## Keyboard Conventions

- CTRL-*x* means to press and hold down the Control key and then press another key on the keyboard, represented here by *x*. This is also used for other key-combinations such as ALT-*x* or Shift-*x*.

- “Type” in a command procedure means to type the command on the command line and then “press” the Enter key.
- “Select” means to choose an operation on a drop-down menu or list.
- “Click” means to click the left mouse button, usually in response to a dialog box. “Right-click” means to click the right mouse button.

## Console Conventions

Commands that you enter at the console, console screen displays, and console prompts are presented in a *typewriter-style* typeface called *Courier*:

- Commands that you need to type are in **Bold Courier**. For example, if you are instructed to type a console command, the instructions may appear as follows:

Type **so** at the `login:` prompt.

- Output to the console screen is in *plain Courier*:

```
AVSTAR-A: list s
T11 miller A
T23 stevens A
T82 allen B
```

Lengthy console displays may be edited to emphasize only the most important information. An ellipsis (...) represents portions of the console display not shown in the text.

The console can display each server’s prompt based on the system ID (typically a station’s call letters) and the server’s name. Examples in this manual use a fictional station and system ID, AVSTAR. For instance, the following is the console prompt for server A on the AVSTAR system:

```
AVSTAR-A:
```

---

## If You Need Help...

### ...In Performing a System Operation

If you are having trouble performing a system operation, you should:

1. Repeat the procedure, carefully following the instructions provided for the task in this guide.
2. Refer to the documentation included with your hardware to review the maintenance procedures or the hardware-related issues.
3. Check the Support section of iNews' Web site at <http://www.inewsroom.com> for online technical publications and additional telephone support phone numbers.
4. Check iNews' Web Bulletin Board, at <http://support.inewsroom.com/~avstar> for information about product and user conferences. If you do not find the answer to your question, you can exchange information with other Avstar customers and iNews Customer Support representatives.
5. Maintenance Agreement contract customers can contact iNews' Customer Support personnel at:
  - 1-800-869-7009 in the USA
  - +44-1256-814222 in Europe
  - +61-2-8877-6888 in Asia/Pacific
  - e-mail [support@inewsroom.com](mailto:support@inewsroom.com)

### ...With the Syntax of Console Commands

If you are at the console, and are unsure about the function of a console command, use the `help` command.

To view instructions about using a command, type `help` followed by the name of the command. For instance, type `help dbvisit` for an explanation of the `dbvisit` command. The following data appears:

```
dbvisit -<d or v> -{r or m name} -[s] [block# ...]
'r' for read only
's' for "slow" to eliminate cache usage
'm' for machine name to disconnect
'i' to just validate isam files
```



*Because of the margin limitations of this manual, console command lines may appear wrapped to multiple lines. This does not necessarily indicate the need to press an Enter key. Unless otherwise indicated, console commands should be typed on a single line, allowing the computer to wrap the text whenever the command line stretches beyond the screen margin.*

## ...With UNIX, or Specific Devices

Your best source for more detailed information about UNIX is the UNIX documentation for your operating system. Any UNIX features not mentioned in this manual are not supported in the Avstar system.

For more information about any device connected to your Avstar system, refer to the documentation included with the device.



---

## Other Documentation

The following documents provide more information pertaining to iNews™ products.

### Avstar Newsroom Computer System Documentation

- *Avstar Newsroom Computer System Installation Manual for SCO and SGI Systems* describes the installation process for customers not now using Avstar Newsroom Computer System.
- *Avstar Newsroom Computer System Update Manual for SCO and SGI Systems* describes the process for updating from an Avid NetStation™ system to Avstar Newsroom Computer System.
- *Avstar Newsroom Computer System Update Manual for the DEC/MIPS System* describes the process for updating from an Avid NetStation system to Avstar Newsroom Computer System.
- *Avstar Newsroom Computer System Release Notes* provides installation, administration, and user-level information that may not have been available at the time the other documentation was printed.
- *Avstar Newsroom Computer System Introduction to Avstar Workstation Training Guide* provides basic user-level information.
- Avstar Newsroom Computer System online help gives you quick-reference information about user-level software functions.

### Broadcast Control System Documentation

- Broadcast Control System (BCS) online help describes the user-level software functions for the Broadcast Control System.
- *Broadcast Control System Operations Manual* provides Avstar system administrators with operational and maintenance information about BCS.

- *Broadcast Control System Release Notes* provides installation, administration, and user-level information that may not have been available at the time the other documentation was printed.

## Other Products

- Contact your iNews Sales Representative for documentation and information on other iNews™ products, such as Media Browse, EditStar®, LeaderPlus™, NewStar®, and so forth.

## If You Have Documentation Comments

We continuously seek to improve our customer documentation. We value your comments about this manual or other iNews-supplied technical publications. That is why we include a Reader's Comments form at the back of this manual. You can fill it out and mail it to the address provided on the form, or you can send your documentation comments by e-mail to the iNews' Technical Publications department at: [TechPubs@inewsroom.com](mailto:TechPubs@inewsroom.com)

Please include the publication title, part number, revision letter (if any), all of which can be found at the bottom of the copyright page in this manual. Also include the specific section that you are commenting on in all correspondence.

# SECTION I

## *Avstar Overview and System Basics*

This section introduces the Avstar Newsroom Computer System. The section consists of the following chapters:

- Chapter 1, *Introduction*
- Chapter 2, *The Avstar Console*
- Chapter 3, *Getting Started*
- Chapter 4, *Users*
- Chapter 5, *Stories, Queues, and Directories*
- Chapter 6, *Groups*
- Chapter 7, *Keyboard Macros*
- Chapter 8, *Forms*

---

# CHAPTER 1

## *Introduction*

Avstar Newsroom Computer System is an integrated, digital news creation and production system. It provides journalists, producers, directors, writers, and technical personnel in a newsroom with an array of tools to make their jobs easier.

This chapter contains:

- What is Avstar?
- iNEWS Products
- Links to Other Newsroom Products
- System Administrator Tasks

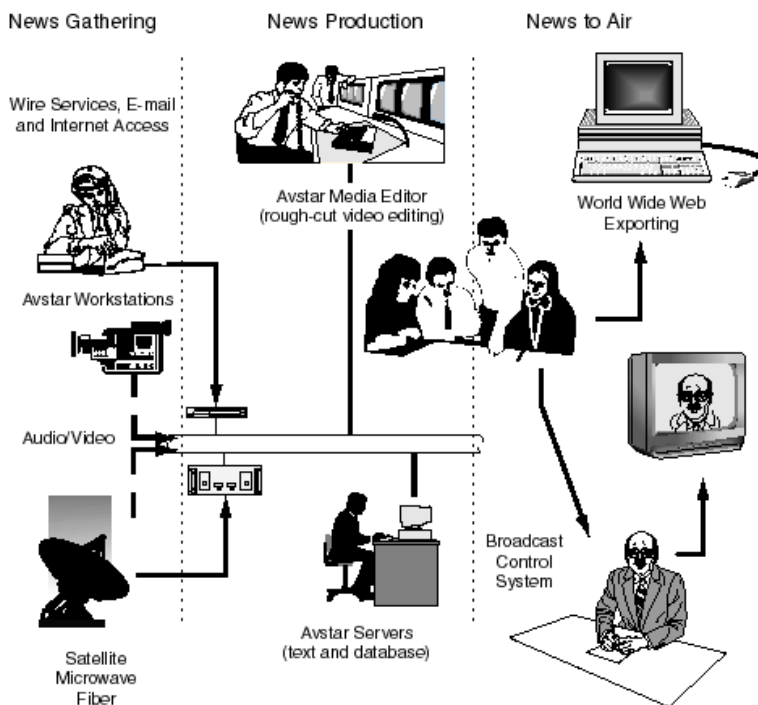
## What is Avstar?

The Avstar system consists of three iNews products: the Avstar Newsroom Computer System, the Avstar Media System, and the Avstar Broadcast Control System. Your newsroom may have any one or all of these products.

An Avstar Newsroom Computer System (NRCS) provides:

- News gathering from video, audio, and text sources
- News production, including:
  - Story creation and script editing
  - Show planning and creation, with media flagging and cutting
- News to air, including:
  - Machine control capabilities for on-air operations
  - File exporting
  - Internet publishing

The figure below shows a typical workflow in an environment with an Avstar system:

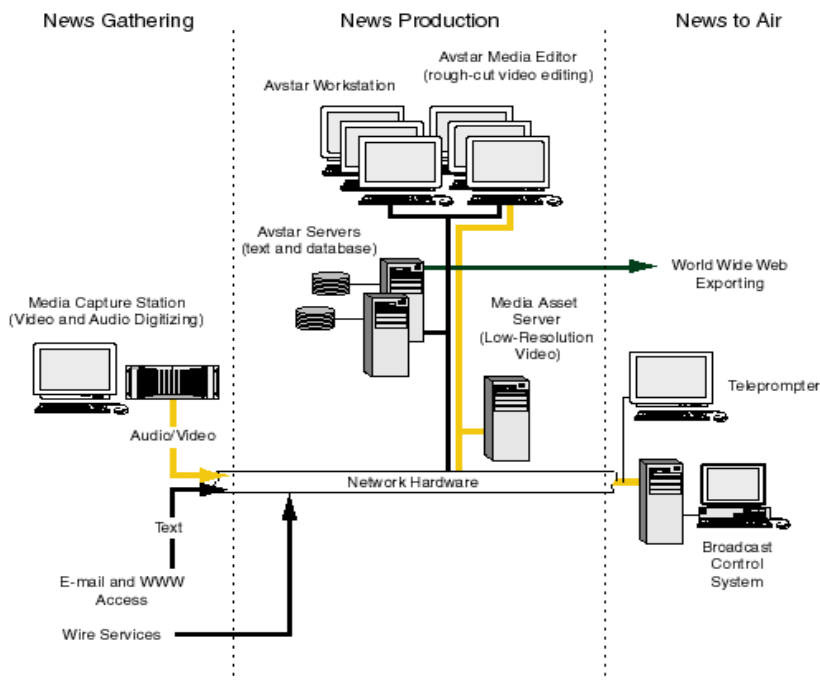


**Figure 1-1 Typical Avstar System Workflow**

## iNEWS Products

This section diagrams and describes the interrelationship between the three iNews products that make up the Avstar system:

- Avstar Newsroom Computer System
- Avstar Media System
- Avstar Broadcast Control System



**Figure 1-2 Interrelationship of iNEWS Products**

## Avstar Newsroom Computer System

The Avstar Newsroom Computer System consists of the Avstar Workstation and Avstar Server components. All components work together as an integrated system and provide journalists, producers, and news directors with an array of digital tools for producing and monitoring shows.

Journalists sitting at Avstar Workstations can simultaneously monitor news wires, work on stories, scan archives, and access an array of online information sources. The Avstar Workstations in your newsroom are linked together via a network so they can share information. When the server receives new information, such as additions or changes to stories, it is immediately available to all newsroom person-



---

nel. Mail and messaging capabilities make group communication fast and efficient.

From their individual workstations, producers and news directors can plan a show and view the progress of a predefined rundown. They can also create rundowns and display timing information for programs. Color highlights on the workstation screen show critical status information, such as overrun, unapproved, and video-ready segments, making it easy to access the status of the show.

Authorized users can log in and access the story database using any standard Web browser. Users can also publish news stories directly to a Web server in Hypertext Markup Language (HTML) with a single command.

The Avstar Server manages all the day-to-day activities of the newsroom. System administrators can create forms-based displays and customize rundowns specific to their newsrooms. The system features a fully mirrored database for immediate cutover in the event of a system failure.

## **iNEWS Media Browse 2000**

The iNews Media Browse 2000 system integrates video production into Avstar Newsroom Computer System, enabling journalists to view and edit low-resolution video on a single workstation. Journalists can view the latest media as it is recorded into the Media Asset Server, marking shots, and annotating and organizing shots into a sequence ready for air or final polish in a high-resolution editing application.

The iNews Media Browse 2000 system removes barriers between newsroom computing and video production. It enables story creation decisions made by journalists to be executed on Avid's NewsCutter®, a high-resolution video editing system. Since the edit decisions are saved as Open Media Framework® (OMF®) files, they can be recreated automatically for on-air delivery. NewsCutter can fine-tune stories further, adding titles and special effects.

For more information about Media Browse 2000, see *Media Browse 2000 Online Help System* or *Media Browse 2000 Operations Manual*.

## **iNEWS Broadcast Control System**

The iNews Broadcast Control System (BCS) is a machine control system for on-air operations. BCS can operate in both integrated and stand-alone operations, and directly controls production and playback devices. It receives information from the Avstar Workstation as control events are entered into scripts, and enables back-to-back show production. If required, it can handle several shows at the same time.

BCS consists of a server and client graphical user interface (GUI). Technical directors have their own Windows-based GUI client to control events on the BCS server.

For more information about iNews Broadcast Control System, see the *Broadcast Control System Operations Manual*.

## **Links to Other Newsroom Products**

The Avstar product set is constructed on a modular, open architecture, enabling its components to work efficiently not only with other iNews products but also with third-party hardware and software. Avstar NRCS provides efficient links to other iNews products for additional scripting and video capabilities. One example is Digital News Gathering (DNG), a disk-based production system that stores digital video, audio, and graphics data in a single central library accessible by client workstations for recording, editing, and playback.

All iNews products operate on industry-standard technologies, including Windows 95 and Windows NT operating systems, Intel-based PCs, Intel and SGI servers, and TCP/IP Internet networking protocols. Also, iNews is publishing open Applications Programming Interfaces (APIs) for the Broadcast Control System that will enable integration with third-party software and hardware.

Figure 1-3 shows a sample layout of an Avstar Newsroom Computer System.

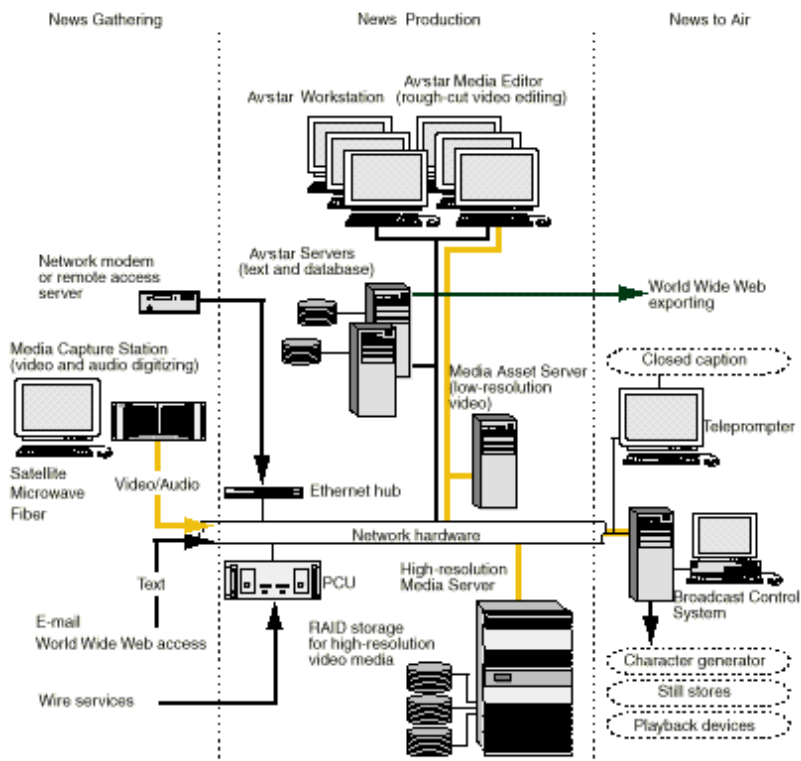


Figure 1-3 Sample Avstar NRCS Layout

# System Administrator Tasks

The following sections introduce many system administrator responsibilities and the system's capabilities and functions.

## Basic System Administration Tasks

Before you can customize or maintain the Avstar Newsroom Computer System, you must learn several basic tasks, which include:

- Start up or shut down Avstar NRCS, which includes logging out users and taking the system offline.
- Backing up a site file before making file modifications.
- Send system administrator commands from the console to one or more of your system's computers.
- Be a console superuser, capable of setting up special superuser permissions.

## User Tasks

A *user* is anyone who can log in to the database and use Avstar NRCS. Your responsibilities regarding users are:

- Monitor user information, such as users' access privileges and which users are currently logged in.
- Customize the traits of users' accounts to enable users to more effectively use the system.
- Provide a new employee access to the information stored in the Avstar NRCS database by creating a new user account.
- Remove user accounts of former employees to prevent improper access to the Avstar NRCS database.

## Database Tasks

The Avstar system database contains the information your organization needs to function. A system administrator's tasks associated with the database include:

- Design forms (that is, story templates) to display important information about stories in a queue.
- Monitor changes to files and queues in the database.
- Unlock or delete any item in the database, and recover items that were accidentally deleted or corrupted.
- Create new folders or queues in the Avstar system database to meet your organization's expanding needs.
- Remove a directory or queue from the database, if it is no longer used.
- Change the name or traits of an existing directory or queue.
- Assign the mailbox trait to queues for configuring automatic story distribution into and out of queues.

## Security Tasks

There are many ways to ensure the security of your Avstar system. Your responsibilities regarding system security include:

- Monitor and change passwords or force users to change them by setting up system checks and modifications.
- Monitor user login activity to guard against unauthorized use of the Avstar system.
- Assign security to a directory or queue, limiting access to a specific group of users.
- Restrict database access by placing users into security groups based on job roles and need for information.

## Customizing Commands and Messages

Your responsibilities regarding commands and messages include:

- Customize command names, message text, and other items by changing their entries in your system's dictionary files.
- Remove your custom dictionary translations by reverting to the default settings for command names, message text, and other items.

## Storage Maintenance Tasks

You will want to monitor the database regularly to ensure adequate storage. Storage maintenance tasks include:

- Monitor how much free space is available in the database and, if necessary, increase the amount to prevent the system from running out of space.
- Perform preventive database maintenance by periodically running certain utility programs that can find and automatically fix minor problems before they become serious.
- Backup the entire database or portions of it onto a tape, so if necessary, the information can be restored to the database later.
- Make a backup copy of the Avstar system software.
- Make a backup copy of files, such as the site file, on tape any time you make a important changes.

## Device Tasks

A *device* is any kind of hardware or software that performs a specific function when it is set up on the Avstar system. Your responsibilities regarding devices include:

- List the parameters of any device running on your system or list all devices of one type.
- Add any type of device to your system, if you have the capacity and license permission.

- Use the UNIX line editor, known as Ed, to change the setup information for a device in your system's configuration file.
- Reconfigure the system so it recognizes any changes you make to your system's devices.
- Change each printer on your system so each has its own set of printing profile options.
- Setup servers which are utility programs that automatically perform various actions on the stories in your database.
- Change wire distribution and sorting of data coming into your database from a wire service to queues based on their category codes or content.
- Write dialogs (lists of instructions) for each service to automate the connection process. A service is a device that connects a user to a remote computer system.
- Design and assign custom keyboards for users with a unique set of keyboard macros.

## Reviewing Default Settings

Your responsibilities regarding system profiles, default settings, and command syntax include:

- Changing a system profile setting to change your system's operation.
- Reviewing default settings of all system profile parameters.
- Reviewing command syntax for edit, console, and job list commands.

## Troubleshooting

Your troubleshooting responsibilities include:

- Transfer system activities from a halted computer to other system computers. If a computer connected to the system has been halted,

bring the system back to operation using the remaining computers.

- Reconnect a computer that has been halted. Following routine maintenance, reintegrate a computer into your system's operation.



# CHAPTER 2

## *The Avstar Console*

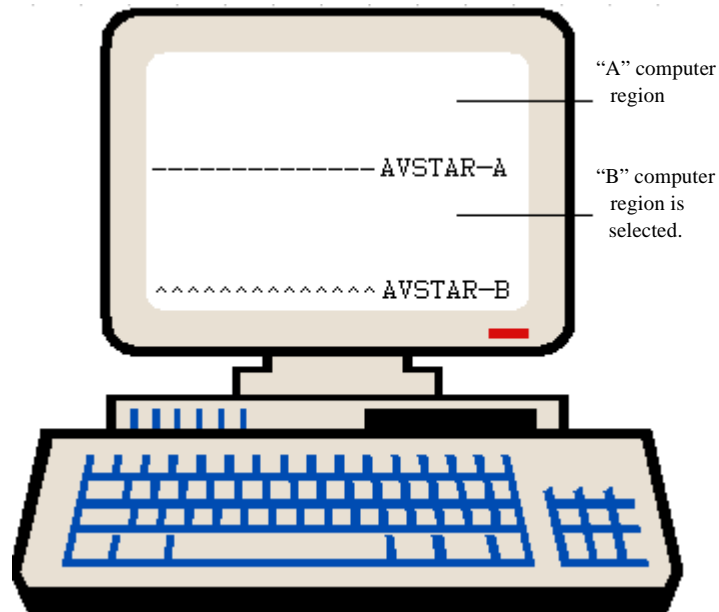
The Avstar console is an IBM/Intel-compatible personal computer (PC) running custom-created Avstar software. The console serves as a “command center” that enables you to monitor and maintain your Avstar Newsroom Computer System.

This chapter contains information about:

- Overview
- Commands You Can Type at the Console
- Selecting Servers
- Console History
- Console Function Keys
- Console Operations
- The Remote Console
- The Console Configuration File (console.cfg)
- Console Control Command Reference

## Overview

Although the console can control multiple computers, known as servers, your console has one screen, which is often divided into regions to separate the output from each server. Figure 2-1 shows a console in a dual-server system—the console screen is divided into two regions. Your console screen has as many regions as there are host servers in your system.



**Figure 2-1 Console Screen for a Dual-Server System**

To identify which region belongs to which server, the console displays the name of the server that a region represents in that region's lower right corner. In the example shown in Figure 2-1, the top region displays the output from server A, and the bottom region displays the output from server B. The wavy lines (^^^) to the left of the identifier for server B indicate that it is currently selected.

## Commands You Can Type at the Console

You can type two kinds of commands at the console:

- *Server commands* are sent to the Avstar servers. For instance, the `list s` command sent to an Avstar server will return information about who is logged in.

These commands are all explained in other chapters of this manual and are summarized in Appendix A, “Command References.”

- *Console control commands* are sent to the Avstar console software that communicates with the servers.

Some common console control commands are explained in the following section. For a complete list and description, see Table 2-4 on page 2-21.

### Console Control Commands

To type console control commands, first press the Enter key on the numeric keypad in the lower right corner of the keyboard.

The Enter key on the numeric keypad will be called the Command key throughout this manual to prevent confusion with the Enter key on the keyboard.

The Enter key on the numeric keypad is the console’s Command (CMD) key. When you press it, the console displays the `COMMAND` prompt—also called the *command line*—from which you can type in commands. After you type a command, you need to press the Enter key—also called the Return key— on the standard keyboard for the command to be executed. Figure 2-2 shows the location of the two keys on a typical Avstar console keyboard.

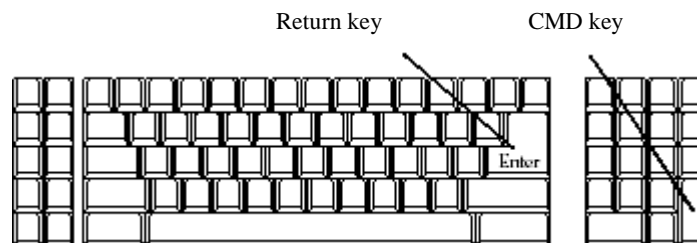


Figure 2-2 Typical Avstar Console Keyboard

### Example: The Computer Command

A typical example is the following sequence that will switch the computer selection in Figure 2-1 from server B to server A:

1. Press the Command key.
2. Type one of the following:
  - a. `computer a`
  - OR-
  - b. `c a`
3. Press the Enter key.

You might follow the previous sequence with server commands to server A, which you would type after the server prompt; it looks something like this:

```
AVSTAR-A:
```

To type another console control command, again press the Command key.

If you make a mistake when typing a command, use the Backspace key or move the cursor back and then type over it. You can also cancel the entire line and start over by typing the “at” character (@)—press Shift-2.

If you need to stop a command, press Delete. If that does not work, hold down the Control (CTRL) key and press the Backslash (\) key. This action stops the command’s execution and displays the `login:` prompt. Type `so` at the prompt to log in again.



*On SGI systems, you might have to press CTRL-D several times to regain normal console function.*

If the server sends a message while you are typing a command, the console stops displaying your keystrokes to display the message. However, it continues to record what you type. After it has displayed the message, the console displays the data that you typed while the message was being displayed.



*If you are interrupted by a console display or have mistyped a line, type the “at” character (@) to cancel what you have typed.*

## Selecting Servers

On the console, you can select one server or multiple servers at the same time. For instance, some commands must be executed on all servers at the same time, so on a two-server system, you would have to select both server A and B before typing in the command. The previous example of the Computer Command showed you how to select server A. Here are a few additional examples.

### Selecting One or More Servers

To select only server B:

1. Press the Command key.
2. Type one of the following:
  - a. `computer b`

-OR-

  - b. `c b`
3. Press the Enter key.

To select both the A and B servers:

1. Press the Command key.
2. Type one of the following:
  - a. `computer ab`

-OR-

  - b. `c ab`
3. Press the Enter key.

To select all servers in the Avstar system:

1. Press the Command key.
2. Type `c *`.
3. Press the Enter key.

Selecting all servers enables you to send a command to all of them simultaneously. When you select all servers, each server region's bottom line changes to a row of ^ characters. Only one cursor appears, usually in the top region. However, the command information you type appears simultaneously in each region of the console screen.

### Zooming in on One Server

In addition to the `computer` command, you can also use the `zoom` command to select a server. Unlike the `computer` command, which operates in *split screen mode*, `zoom` selects one server at a time and devotes the entire console screen to that server.

Figure 2-3 shows the console screen after “zooming in” on server B's region.



Figure 2-3 “Zooming In” On Server B

To zoom in on one server, such as server B:

1. Press the Command key.
2. Type one of the following:
  - a. `zoom b`
  - OR-
  - b. `z b`
3. Press Enter.

To restore the screen to its former split-screen state, use the `computer` command to select any server. It does not matter which one you select. For instance, type `c a` for server A. The console screen will display multiple regions, and the region for server A will be selected.

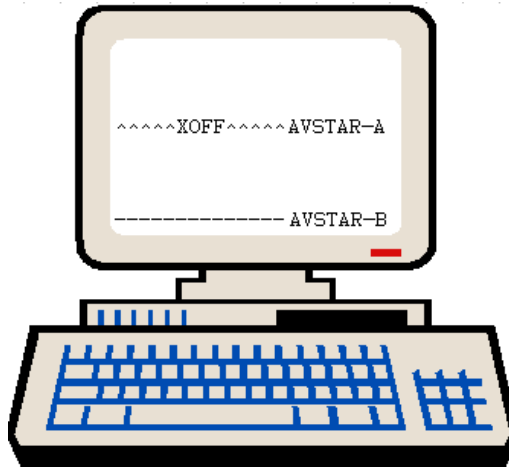
## Console History

Relatively short system responses to your commands appear on the console screen below your command text.

Longer output, however, sometimes scrolls off the screen. You might want to go back and view this in screen-sized chunks; you can do this by pausing the screen display.

## Pausing the Screen Display

To pause the screen display, press CTRL-S, which temporarily stops the console screen from scrolling. When the screen is full of text, XOFF appears at the bottom of the selected region, as shown in Figure 2-4.



**Figure 2-4** XOFF Message Indicating Scrolling Is Paused

To resume scrolling, press CTRL-Q. Even if you do not press CTRL-Q, the console automatically resumes scrolling after a pause of 60 seconds.



*Pressing CTRL-S does not have any effect if you are using a remote console. A remote console is connected to the network by a modem dial-in from an external location. See “The Remote Console” on page 2-14 for more information.*

## Viewing Recent Console History

The console maintains a history buffer containing messages that have appeared on the screen. You can go back in “console history” to review prior activity on any of the Avstar NRCS servers. The “top” of the buffer contains the oldest information; the “bottom” of the buffer contains the most current.



To view recent history on a particular server:

1. Select and zoom in to the server whose history you want to review.
2. Press the Command key.
3. Type one of the console history commands: **up**, **down**, **top**, or **bottom**.
4. Press Enter.

Table 2-1 shows examples of console history commands.

**Table 2-1 History Commands**

Console History Command	Has the following effect:
<code>up monitor</code>	Goes up in history searching for the word "monitor."
<code>up 180</code>	Goes up 180 lines from the current line.
<code>down 50</code>	Goes down 50 lines from the current line.
<code>top</code>	Goes to the top of console history.
<code>bottom</code>	Goes to the bottom of console history.

When you are in console history, the console prompt changes to `History` to indicate you are in history viewing mode and the console is ready to accept another history viewing command.

Once you have found the section of the console history you are looking for, you can "print" a section to the screen as new console output. For instance, after you have gone up 180 lines by pressing the Command key and typing `up 180`, you might want to print the next 20 lines. To do that, type `pr 20`.

To print all lines, from your current location in the history to the most recent console activity, type:

`pr all`.

## Reading Older History

You can configure Avstar to log console history to disk for later review. The logs are written to the hard drive on your console PC, traditionally in the `C:\Console` directory. The logs are named as shown in Table 2-2.

**Table 2-2 Log Names**

File for server A:	File for server B:	Contains the following information:
<code>log.a1</code>	<code>log.b1</code>	Most recent history
<code>log.a2</code>	<code>log.b2</code>	Old history
<code>log.a3</code>	<code>log.b3</code>	Older history
<code>log.a4</code>	<code>log.b4</code>	Oldest history

The `log.a1` file is a duplicate of what is in current history; you can also view it as shown in prior examples using the `up` command.

As the `log.a1` file fills up, old files are renamed and a new `log.a1` file is created as follows:

- `log.a1` is renamed `log.a2`
- `log.a2` is renamed `log.a3`
- `log.a3` is renamed `log.a4`
- A new `log.a1` is created

The log files are ASCII text files that can be read with any word processing program. You must exit the Console program if you want to edit the logs in any way.



*You can use the console `view` command to view the log files without editing or exiting to DOS, but `view` only lets you start at the top of the file and scroll down. You cannot move back or search for words.*

Both the presence or absence of disk logging and the size of the log files can be configured in the `CONSOLE.CFG` file. See “The Console Configuration File (`console.cfg`)” on page 2-16 for more information.

## Console Function Keys

You can preprogram your keyboard function keys to execute commands. For instance, you might program F1 to select server A, F2 to select server B, and F7 to move up 200 lines in the console history buffer.

### Assigning a Command to a Function Key

To assign a command to a function key:

1. Press the Command key.
2. Type the name of the function key you want to use, followed by the equal sign (=) and the command the key is supposed to execute. The following example assigns the command of choosing server A to the F1 key: `f1=c a`.
3. Press Enter.

To assign a command sequence to a function key—that is, include the Command and Enter keys in the definition—use the open brace ({} to represent Command, and the close brace (}) to represent Enter. For example, to program the complete command sequence (press Command key, type computer command to select all servers, and press Enter) to function key F10, you would type: `f10={c *}`

### Changing the Assignment of a Function Key

To change a command assigned to a function key, assign a new definition to the key.

### Deleting the Definition of a Function Key

To delete a function key’s command assignment, assign it a null value.

## Displaying Function Key Assignments

To find out the command (if any) assigned to a key:

1. Press the Command key.
2. Type the name of the key, such as, **F9**.
3. Press Enter.

Press Command again to clear the command assignment from the console screen.

## Console Operations

This section explains what to do if the console freezes and how to start and exit the console.

### If the Console Freezes . . .

If the servers on your system are not responding to commands and are not displaying messages:

1. Check to make sure that you or someone else has not stopped scrolling. (If that is the case, **XOFF** is displayed under the region where scrolling has stopped.)
2. Type **CTRL-Q** to start scrolling.

It may also be possible that the server ports have stopped sending and receiving. To start that activity again:

1. Press the Command key.
2. Type **x**.
3. Press Enter.

If this does not restore the console, static electricity may have frozen one or more of the servers' I/O ports. You may be able to unfreeze these ports using the `reset` command. To do that:

1. Select the affected server(s).

2. Press the Command key.
3. Type `r` (for `reset`).

Another possible cause of a frozen console is an application program that will not stop running. If you suspect this problem:

1. Select the affected server(s).
2. Hold down the Control and Shift keys, and type a backslash (\).

This stops the program on the selected servers and causes them to display the `login:` prompt.

3. Press Enter and log in as system operator.

See “Logging in As System Operator” on page 3-2 for more information.



*On SGI systems, you might have to press CTRL-D several times to regain normal console function. If you press CTRL-D one too many times, you might log out as system operator and have to log back in again, as explained in “Logging in As System Operator” on page 3-2.*

If the console still does not respond, exit the console program and restart it as described in the following sections.

## Exiting the Console

You should normally leave the console on at all times while the Avstar system is running. However, the following situations may require you to exit the console:

- The console is frozen, and you are unable to unfreeze it using the methods described in “If the Console Freezes . . .” on page 2-12.
- You need to change the console’s configuration file. See “The Console Configuration File (`console.cfg`)” on page 2-16 for more information.

To exit the console:

1. Press the Command key.

2. Type **CTRL-e**.
3. When `COMMAND EXIT` appears, press Enter.

Once the console software completely shuts down, the prompt for your operating system will appear.

## Starting the Console

If your console has been turned completely off, it should start the console program automatically when it boots up.

However, if you are starting up the console from your operating system prompt, do the following:

1. Type **console**.
2. Press Enter.

## The Remote Console

The Avstar console can have a modem attached to it. This enables someone in another location to call up the console and log in, thereby turning the remote computer into a *remote console*. The primary use for a remote console is to enable technicians or system administrators to perform diagnostic and maintenance work on the Avstar system from a remote location.

This sections explains how to dial in over a modem line, what you can expect to see on the remote screen, and how to execute commands remotely.

## Dialing in to the Console

To dial in to the console, you must have a terminal or computer that transmits and receives ASCII characters. You must also have set the following modem options:

- Eight data bits
- No Parity

- One stop bit
- Any baud rate supported by your console's modem

To prevent unauthorized people from dialing in to the console, remote access is protected with a password. When you dial in, you see a `PASSWORD` prompt on the screen. (If you do not see the prompt immediately, pressing Enter should display it.) After you type the correct password, the console connects you to the first server listed in the console configuration file, usually server A.

At the console, MDM is displayed at the bottom of the region representing the server that was selected from the remote console. Commands typed at the remote console are sent to that server and displayed on its console region. Likewise, commands sent by that server are displayed both on the console and the remote console.

## Executing Commands Remotely

Once logged in, you can type commands and review history almost as if you were seated at the console itself. All commands except `zoom` and `history` are available from the remote console. Just as at the console, these commands can be abbreviated using the first letter in each command.

There are also some differences when using a remote console:

- The remote console displays screen input and output for only one server at a time, even if you have more than one selected.

When you select two or more servers, the order in which you list the servers in the computer command determines which server's display you see. For instance, if you type `c ba` to select servers B and A, you see output only from server B on the remote, even though what you type is sent to both A and B.

- Use the Escape key instead of the Command key to display the command prompt.

## Logging out from a Remote Console

When you have finished using the remote console, log out by doing the following:

1. Press the Escape key.
2. Type `l` (for `logout`).
3. Press the Enter key.
4. When the remote console redisplay the `PASSWORD` prompt, hang up your modem.

## Logging out a Remote User from the Main Console

If you are at the main console and discover that you or someone else has been using the remote console but did not log out when they were done, you can log them out by doing the following:

1. Press the Command key.
2. Type `m` (for `modem`).
3. Press the Enter key.



*Always follow this procedure before disconnecting the modem on the main console.*

## The Console Configuration File (`console.cfg`)

The Avstar console uses information in a configuration file (a text file called `console.cfg`) to set a number of parameters, such as:

- Whether or not disk logging is enabled
- Information about each of the servers connected to the console
- Information about the remote console

This section shows a sample configuration, and defines the console configuration keywords and their parameters.



## Looking at the Console Configuration File

To view the configuration file:

1. Zoom to display only one region on the screen. For example:
  - a. Press the Command key.
  - b. Type **z a**.
  - c. Press Enter.
2. Use the `view` command to view the configuration file:
  - a. Press the Command key.
  - b. Type **v console.cfg**.
3. Press Enter to display the first line in the console configuration file.
4. Continue pressing Enter to scroll through the file.

Here is a sample console configuration file:

```
log b:log
computer           ;"A" server
name a
label AVSTAR
irq 3
hostess 2c0
speed 1200
;
computer           ;"B" server
name b
label AVSTAR
irq 3
hostess 2c8
speed 1200
;
```

```
computer                ;"C" server
name c
label ARCHIVE
irq 3
hostess 2d0
speed 9600
;
modem                    ;Remote console
password turtle
irq 3
hostess 2d8
speed 1200
```

As you can see, this file consists of a list of keywords (such as, `name` and `label`), most of which are followed by parameters (such as, `a` and `AVSTAR`). The keywords are described in detail in Table 2-3, “Console Configuration Keywords” on page 2-19.

The keywords `modem` and `computer` identify the start of the modem (remote console) section and the server sections. Each server (including the archive server, if your system has one) and the modem must have their own sections in the console configuration file.

## Editing the Configuration File

Probably the only modification you will ever need to make to the console’s configuration file is to change the modem password.



*The console configuration file is stored on a DOS PC, so you need to use a DOS editing tool, such as `edit` to change it. You could also copy the file to a diskette and take it to another location to edit using Microsoft’s program, `NOTEPAD`, on a Windows-based PC.*

## Console Configuration Keywords

Table 2-3 lists all the configuration keywords and their parameters, if they have any.

**Table 2-3 Console Configuration Keywords**

Keyword	Explanation
<code>computer</code>	Indicates the beginning of a server section. Must appear at the top of each server section on the configuration file.
<code>hostess &lt;port-address&gt;</code>	Indicates which port address (in hex) the console uses to communicate with a particular device (that is, a server or the modem). This information, which is dictated by the hardware, was placed in the configuration file when your system was installed and should not be changed.
<code>irq &lt;interrupt-request-number&gt;</code>	<p>To get the attention of the console, each device (that is, the servers and the modem) connected to the console must have its own <i>interrupt request</i> with which it can signal the console. The <code>irq</code> keyword tells the console which interrupt request to expect from each device.</p> <p>This information, which is dictated by the hardware, was placed in the configuration file when your system was installed and should not be changed.</p>
<code>label &lt;region-name&gt;</code>	Defines a label that the console uses to identify each server's region of the console screen. The label can be up to 15 alphanumeric characters long.
<code>log &lt;drive:filename&gt;</code> <code>[server(s)] [max size]</code>	<p>Enables disk logging. No matter what filename (for example, <code>history</code>) you designate here, the system always uses extensions like A1, A2, and B1, as indicated in "Reading Older History" on page 2-10. If you do not follow the filename with a list of the servers for which you want history to be recorded, the console records history for all servers. If you omit a log size it defaults to 16,384 bytes.</p> <p>This example creates a history file on server B for servers A and B:</p> <pre><b>log b:history ab</b></pre>

**Table 2-3 Console Configuration Keywords (Continued)**

<b>Keyword</b>	<b>Explanation</b>
<code>modem</code>	Indicates the beginning of a modem section of the configuration file.
<code>name &lt;computer-name&gt;</code>	Names the server described in that section of the configuration file. Each server must have A, B, C, or D as its name.
<code>password &lt;password&gt;</code>	The modem password that must be typed when someone logs in at a remote console. The password can have up to eight alphanumeric characters.
<code>portaddress &lt;port&gt;</code>	Selects the DOS address (in hex) the console should use to communicate with a particular device, such as a modem. This information which is dictated by the hardware, is placed in the configuration file during installation and should not be changed. Each device must have a port address defined in its section. The <code>portaddress</code> keyword is used instead of <code>host-ess</code> when your system has five servers or four servers and a modem. In that case, the fifth server or the modem must use <code>com1</code> as its port. The <code>port</code> parameter should always be defined as <code>3f8</code> .
<code>speed &lt;baud-rate&gt;</code>	Sets the baud rate for communication between the console and the server modems. The baud rate is the only communication parameter you can alter; the console ports always communicate at 9600 baud, eight data bits, no parity, and one stop bit.
<code>timeout &lt;minutes:seconds&gt;</code>	<p>Allows you to set a time-out value for any modem connection. The system automatically logs out a modem connection if there is no activity for a specified amount of time. For instance, a value of <code>6:00</code> would automatically log out a modem connection after six minutes of inactivity. This keyword, which should only be used in the modem section, provides added protection should a user forget to logout from a modem connection to the console.</p> <p>With a value of <code>0:00</code>—the default value—the feature is disabled, which means the system will not log out a modem connection regardless of inactivity length. The maximum value is 546 minutes and 7 seconds (<code>546:7</code>).</p>

## Console Control Command Reference

Table 2-4 lists available console control commands and their functions.

**Table 2-4 Control Command Reference**

Keyword	Explanation
<code>bottom</code>	Moves you to the newest (bottom) line in the history.
<code>computer [computer-name(s)   *]</code>	Selects one or more of your system's servers, so that you can type a command on the selected server(s). Follow the command with the name(s) of the servers you want to select, or type an asterisk (*) to select all servers, such as <code>computer *</code> .
<code>down &lt;number-of-lines&gt;</code>	Moves you that many lines forward in the history. For example, typing <code>down 10</code> takes you 10 lines forward. If you follow <code>down</code> with a number greater than the number of lines between your current position and the last line in the history, <code>down</code> moves you to the last line in the history.
<code>down &lt;keyword&gt;</code>	Causes a search for the keyword from your current position forward. For example, <code>down list</code> moves forward to a line containing the word "list." The command <code>down</code> by itself moves you down one line. The <code>down</code> command is not case sensitive. If you specify <code>PEOPLE</code> , <code>down</code> considers <code>people</code> to be a valid match. If <code>down</code> does not find the keyword before reaching the bottom of the history, the console beeps and you are returned to the current line.  A wildcard character (#) can be used to match any character or to a search for a number. For instance, <code>down 160#</code> will search for the number 160 instead of moving down 160 lines.
Exit Function	To leave the Avstar console program and return to the MS-DOS prompt: <ol style="list-style-type: none"> <li>1. Press the Command key.</li> <li>2. Press CTRL-E.</li> <li>3. Press Enter.</li> </ol> <p>(You cannot type <code>exit</code> at the Command prompt.)</p> <p>Type <code>console</code> at the prompt to restart the console program.</p>

**Table 2-4 Control Command Reference (Continued)**

Keyword	Explanation
<i>function-key-number=definition</i>	<p>Assigns a command to a console function key. For instance, to assign the <code>list s</code> command to the F9 key, type <code>f9=list s</code>. To include Command and Enter keys in a function key definition, use braces. For instance, to include those keystrokes in the previous assignment example, type <code>f9={list s}</code>. To list the current assignment of a function key, type the key number by itself on the command line. The valid range is <code>f1-f14</code>. It is recommend no definition be made for F14, which defaults to the Command key. F13 corresponds to the plus (+) key on the numerical keypad.</p>
<code>list [#-of-lines   all]</code>	<p>Sends some number of lines of the history to the printer.</p> <ul style="list-style-type: none"> <li>• <code>list</code> followed by a number, such as <code>list 3</code>, prints that many lines of the history beginning at the current line.</li> <li>• <code>list all</code> prints everything from the current line to the newest line.</li> <li>• <code>list</code> with no parameter prints the current line.</li> </ul> <p>When using <code>list</code>, the word <code>PRINTER</code> appears on the command line. If <code>PRINTER</code> is displayed but nothing is being printed, ensure that the printer is plugged in, turned on, online, and has paper. The console assumes a printer is connected to the PC's parallel port, where output is sent. To cancel a <code>list</code> command while output is printing, press any key; printing stops and your position in the history moves to the last line sent to the printer</p>
<code>logclose</code>	<p>Writes all history currently in memory to disk and then disables disk logging. You can use it to change log disks.</p>
<code>logopen</code>	<p>Resumes history disk logging after it has been suspended with <code>logclose</code>.</p>
<code>logout</code>	<p>Logs you out from a remote console. To log out, press the Escape key and type <code>logout</code>. The remote console displays the <code>PASSWORD</code> prompt, and you can then hang up your modem.</p>
<code>modem</code>	<p>Typed at the main console, this command logs out a remote console user. Before you type this command, make sure that the remote console user is not in the middle of an operation.</p>

**Table 2-4 Control Command Reference (Continued)**

Keyword	Explanation
<pre>print [number-of-lines   all]</pre>	<p>Displays a number of lines of the history on the console screen.</p> <p><code>print</code> followed by a number displays that many lines of the history beginning at the current line.</p> <p><code>print all</code> displays everything from the current line to the newest line.</p> <p><code>print</code> with no parameter displays the current line.</p> <p>To cancel a <code>print</code> command while it is displaying console history, press any key. The console stops at the line last displayed on the screen, and your position in the history moves to that line.</p>
<pre>reset</pre>	<p>Attempts to unfreeze one or more of the console's I/O ports, if communication has failed between the console and your system's servers.</p> <p>To use <code>reset</code>, select the servers that are affected, press the Command key, and type the command.</p>
<pre>top</pre>	<p>Moves you to the oldest line in recent console history. To see history older than this, use the <code>view</code> command to view a history log file on disk.</p>
<pre>up [number-of-lines   keyword]</pre>	<p>Moves you backward (up) some number of lines in the history.</p> <p><code>up number-of-lines</code> moves you that many lines back in the history. For example, typing <code>up 30</code> moves you back 30 lines.</p> <p><code>up keyword</code> searches backward through the history from your current position for that word. For example, to search backward for a line containing <code>dbpurge</code>, type <code>up dbpurge</code>.</p> <p><code>up</code> with no parameter moves you back one line.</p> <p>The <code>up</code> command is not case sensitive. If you specify <code>PEOPLE</code>, <code>up</code> considers <code>people</code> to be a valid match. If <code>up</code> does not find the keyword before reaching the bottom of the history, the console beeps and you are returned to the current line.</p> <p>A wildcard character (<code>#</code>) can be used to match any character or to a search for a number. For instance, <code>up 160#</code> will search for the number 160 instead of moving up 160 lines.</p>

**Table 2-4 Control Command Reference (Continued)**

Keyword	Explanation
<code>view &lt;drive:filename&gt;</code>	<p>Displays a DOS text file on your console screen.</p> <p>Use this command to look at old history that has been saved to disk or to read other disk files, such as the console configuration file. Before you type this command, use the <code>zoom</code> command to display only one server's region on the console screen. (You can choose any server.)</p> <p>While you are viewing a file, you can only move down through it (as opposed to back or up in the file). Each time you press the Enter key, the file scrolls down one line.</p> <p>To stop viewing the file and return to normal console operation, press the Command key.</p>
<code>x</code>	<p>Restarts the sending and receiving of information by the computers' console ports. It stands for XON and causes an XON character to be sent to the server for each selected region. If you are having trouble communicating with your servers from the console, try this command. If it does not work, use the <code>reset</code> command.</p>
<code>zoom &lt;computer-name&gt;</code>	<p>Selects a server and fills the console screen with its region. To return the screen to its normal split-screen state, select any server with the <code>computer</code> command.</p>



# CHAPTER 3

## *Getting Started*

As the Avstar system administrator you will need to log on to Avstar NRCS differently than other users. You will have access to features that others do not. As system administrator, you will be responsible for knowing how to startup and shutdown Avstar NRCS.

This chapter contains information about:

- Logging In As System Operator
- Becoming a Console Superuser
- Changing the System Administration Passwords
- Startup and Shutdown

## Logging in as System Operator

Ordinarily, you are always logged in on each of your system's servers as the system operator.

To log in as the system operator, do the following:

1. When any of your servers displays the `login:` prompt, select that server.
2. Type `so`.
3. Press Enter.
4. If your system has a password for this account (most do), then type in the password when prompted.

For information about creating or changing the system operator password, see “Changing the System Operator Password” on page 3-4.

## Becoming a Console Superuser

A console superuser has special system privileges that allow more powerful (and therefore potentially more dangerous) commands. The console's prompt is the visual indicator for whether you are logged in as a system operator or a superuser. The system operator prompt ends with a colon (:). The superuser prompt ends in a pound sign (#). If a command example in this manual shows the superuser prompt—ending in a pound sign (#)—you must be a superuser to use the command.

The superuser prompt looks like this:

```
AVSTAR-A#
```

The system operator prompt looks like this:

```
AVSTAR-A:
```

Follow the procedures explained in the next section to get to the superuser prompt.

## Entering Superuser Mode

To become a console superuser, you need the console superuser password.

To log in as a console superuser:

1. Type `su`.
2. Type the superuser password at the `password` prompt.

To keep the password confidential, the console does not display what you type.

After you type the password correctly, the console shows that you are a console superuser by changing the colon ( : ) at the end of the console prompt to a pound sign ( # ). If you enter an incorrect password, the console displays an error message and lets you try again.



**To prevent users from typing unauthorized commands, never leave the console unattended when in superuser mode. You should log in as superuser only when you need to type a superuser command, and exit console superuser mode immediately after typing the command.**

## Exiting Console Superuser Mode

To exit from console superuser mode and return to the system operator mode, press CTRL-D. The console shows that you are a console system operator by changing the pound sign ( # ) at the end of the console prompt to a colon ( : ).

# Changing the System Administration Passwords

This section contains information about selecting passwords and changing the passwords for system operator and superuser.

## Selecting Passwords

The system differentiates between uppercase and lowercase characters, so always enter your password in the same case you used when

you created it. The Avstar system operator password must be between six and ten characters and contain at least one numeric character.

Keep a confidential record of changes to system administrator passwords. Knowing the passwords is critical not only for `dbtraits` and other commands, but for iNews Customer Support technicians who may require access for problem diagnosis and reconfiguration. If you forget your passwords, the system may have to be rebooted after all the software is reloaded by iNews Customer Support technicians.

## Changing the System Operator Password

To change the system operator password:

1. When you are logged on as system operator, type:  
`passwd`
2. When prompted, type a new password.



*Systems using a SCO UNIX platform let you choose a password or have the system provide one. Select option 1 to choose your own password. You will be prompted to enter the password.*

3. When prompted to confirm, type the new password a second time.

## Changing the Superuser Password

To change the superuser password:

1. Become a console superuser by typing `su` and the superuser password at the password prompt.
2. Type `passwd root`.
3. When prompted, type a new password.

The UNIX name for superuser is root.



*Systems using a SCO UNIX platform let you choose a password or have the system provide one. Select option 1 to choose your own password. You will be prompted to enter the password.*

4. When prompted, type the new password a second time.

If the password does not match, the system assumes that you made an error and displays an error message. If you see this message, type `passwd root` again.

5. Press CTRL-D to exit from console superuser.

## Startup and Shutdown

The following sections describe procedures for starting up and shutting down the Avstar Newsroom Computer System (NRCS).

### Starting the System

The following procedure shows you how to reboot your servers and synchronize them so that they run together as a single system.

Depending on how you shut down your system, you can begin the procedure in one of two ways:

- If you turned off the servers and they have an autoboot switch, ensure each server's autoboot switch is on. Then turn on the servers as described in Step 1a, in the following procedure.
- If you halted servers when you shut down the system, use the boot command described in Step 1b, in the following procedure.



*Because server types vary, certain displays associated with the startup procedure may also vary. Examples are provided in each case and the type of server is shown in the margin. Because this procedure applies to an entire system that has been shut down, you must perform all the steps on all servers, except where otherwise indicated.*

To start your Avstar system, do the following:

1. Power up or reboot servers in one of the following ways, depending on how you shut down your system:
  - a. If you turned off servers when you shut down your system, boot them to the login prompt by turning them on.

- b. If you halted the servers when you shut down your system, boot each server from the console. Servers that have their operating systems halted display the boot prompt on the console. Select *all* servers using the `C *`  command and enter your server's boot command.

## SGI Systems

For an SGI IRIX system, if you turned off the servers, turn on each server. The console displays: (only partial display shown)

```
Running power-on diagnostics...
```

```
    Starting up the system...
```

```
To perform system maintenance instead, press
<ESC>
```

```
...
```

```
The system is ready.
```

```
Avstar News System
```

```
login:
```

## SCO Systems

For a SCO UNIX system, if you turned off the servers, turn on each server, which then displays a boot prompt, as in the following example:

```
SCO UNIX System V/386 on i0486
```

```
Boot
```

```
:
```

Press Enter to continue. As each server boots, it displays copyright and hardware configuration messages such as these:

```
hd(40)unix systty=sio auto
```

```
Loading kernel hd(40)unix .text
```

```
.....
```

```
Loading kernel hd(40)unix .data
```

```
.....
```

```
Loading kernel hd(40)unix .bss
```

```
...
```



*The system will prompt you to type either the root password for system maintenance or CTRL-D for normal startup to continue booting. It will also display the current system time and prompt you to type in a new time. If the displayed time is correct, you can just press Enter and continue booting.*

```
The system is ready...
```

```
Avstar News System
```

```
login:
```

When each server finishes booting, it displays a `login:` prompt.



**If you did not shut down the system as described, check the console history for messages indicating that all servers shut down at the same time. Do not connect servers unless you are sure they are mirrored. If you cannot find messages indicating simultaneous shutdown, or are otherwise unsure whether the disks are mirrored, call iNews Customer Support for assistance before proceeding.**

**If you shut down the system as instructed, the system mirrors the databases and you can continue the startup procedure.**

2. Select *all* servers. See “Selecting One or More Servers” on page 2-5 for more information.
3. Type `so` and the password, when prompted, to log in as system operator.
4. Type `connect #` to connect.

The `#` character acts as a place holder for each server name. It is replaced with each server’s computer name before the console sends the connect command to all servers in the system. This allows system administrators to send commands to multiple servers without having to select each server and send commands individually. For instance, the `connect #` command sends `connect a` to server A, `connect b` to server B, and so forth.

When connected, each server displays status messages and the system prompt returns.

Messages similar to the following appear:

```
Network interface in0 marked UP address
125.0.0.1 netmask 255.0.0.0
100 aliases longest (alias producer) 15 bytes,
4000 bytes total
A is offline
System is AB. Master is A.
Disk status is OK.
AVSTAR-A:
```

Connecting servers provides each server with a unique name and causes each one to read and interpret the system profile. The servers can work together as a system after reading the system profile information.

5. (Optional) Check for edit and order locks if you are restarting the system after a power failure.

During a power failure, the system may not have had time to remove edit and order locks from the database before shutting down. When you restart the system, remove these locks.



*Checking for edit and order locks may take time depending on the size of the database. In an emergency, bypass this step to get the system running. Go back later and remove locks to provide system access.*

*The system can detect invalid locks and will ignore them.*

To remove edit and order locks, select one server and type:

```
dbclean -x .
```

The `-x` option tells `dbclean` to skip queues or directories marked with a skip flag, reducing the time it takes to run.

The period (`.`) after the `-x` causes `dbclean` to start at the root directory of the database, so that it does not miss any part of the database not marked with a skip flag.

6. Select *all* servers. See “Selecting One or More Servers” on page 2-5 for more information.



## 7. Type **startup** to start the system.

Information similar to the following appears:

```
A Fri Aug 17 17:32:15 msg: System being
configured.
checking free space
data base size (113977) free blocks (1100)
starting news programs
booting pcu 10 on port 1
booting pcu 20 on port 2
```

The **startup** command does the following:

- Causes the master computer (usually server A) to read the configuration file
- Brings each server online so users can log in

The console displays device-ready messages (**Hot-to-go**) as each device starts up, indicating that the device is online and available.



*Resources used for Avstar Workstation sessions do not print any messages until a workstation establishes a connection.*

## Shutting Down the System

If you need to turn off your servers or reboot the system, first shut down the system. Shutting down the system:

- Saves any open stories
- Removes any remaining edit and order locks
- Ensures that each server's copy of the database is the same



Because the system requires that you shut down all servers at the same time, most steps in this procedure are performed on all servers simultaneously. Except where instructed to do otherwise, ensure that you have selected all servers using the `computer` command before performing each step. See “Selecting One or More Servers” on page 2-5 for more information.

To shut down your Avstar Newsroom Computer System, do the following:

1. Select *all* servers and type `offline` to take the system offline.

The `offline` command prevents users from logging in.

2. Select *all* servers and type `broadcast` followed by the message warning users already logged in that the system will be shut down. Include the time the system will be shut down. Here’s an example:

```
AVSTAR-A: broadcast WARNING!System shut down 12PM
```

The `broadcast` command broadcasts a message to all users logged in at present.

3. At the specified shutdown time, select *one* server and type the command `list s` to check who is still logged in.

A message similar to the following appears:

T11	miller	A
T82	allen	B
T101	stevens	A
R801	stevens	A

The `list s` command lists:

- The device controlling the session
- The user account used for the session
- The server servicing the session

4. Select *all* servers.

5. Type `logout all` to log out all users. If a user is editing a story, this saves the file and logs out the user.
6. Type `list s` again to check for connect session users.

The `logout all` console command does not log out users who are currently in a connect session.

```
AVSTAR-A: list s
T101  stevens      A
R801  stevens      A
```

If any users are still logged in, notify them of the shutdown by some other means, such as by telephone.



**If a user is in a connect session when you shut down the system, the user's workstation stops, the session is disconnected, and any unsaved work is lost. Ensure any connect session users have logged out before you continue the shutdown procedure.**

7. Type `shutdown` to shut down the system.

A message similar to the following appears:

```
WARNING! This will stop all devices on this
computer, and close the database.
To prevent loss of work in progress, 'logout
all' first.
Do you really want to do this (y/n)?
```

8. Type `y` to continue:

```
Do you really want to do this (y/n)? y
/exc/shutdown: Stopping all devices
/exc/shutdown: Closing database
```

The shutdown process stops all workstations, wires, and other devices, and no further changes can be made to the database.

9. Type the **su** command at the prompt and the superuser password at the password prompt to become a console superuser:

```
AVSTAR-A: su
password:
SU: so /dev/console
```

#### SGI Systems

On SGI servers, type **sync** to save changes to the system software, and type **halt** to halt the system. You must be a superuser.

```
AVSTAR-A# sync
AVSTAR-A# halt
Shutdown started...
...Running power-on diagnostics...
OK to power off system now
Press any key to restart
```

As shown, the console displays a message when it is ok to continue.

#### SCO Systems

On SCO systems, shut down the system by typing **init 0** (that is, zero) at the prompt. You must be a superuser.

```
AVSTAR-A# init 0
INIT: New run level: 0
The system is coming down. Please wait.
System services are now being stopped.
...
The system is down.
**   Safe to Power Off           **
      -or-
** Press Any Key to Reboot      **
```

10. Turn off each server.

When you are ready to start up your system, follow the procedure described in “Starting the System” on page 3-5.

# CHAPTER 4

## *Users*

People in your newsroom must have user accounts to use the Avstar Newsroom Computer System (NRCS). Each user account has various *user traits* associated with it that capture information about the user's interaction with the system—information such as passwords, keyboard preferences, and permissions for story editing.

This chapter tells how the system administrator can access and change user account information from any Avstar Workstation. However, user traits can also be viewed and modified at the Avstar console. The procedures for using the console is covered in Appendix G, “Managing Traits at Console,” on page G-1.

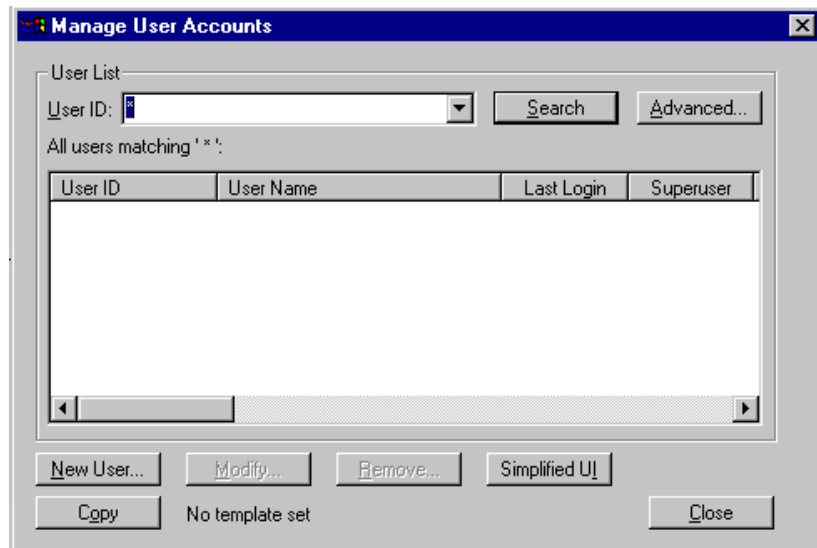
- Viewing User Accounts
- Modifying User Traits
- User Traits Summary
- Simplified User Settings
- Setting up New Users in Avstar
- Searching for Information About Users
- Removing User Accounts
- Creating a User Manager Account
- Creating a Database Manager Account

## Viewing User Accounts

To look at the traits associated with a particular user account:

1. Click on the Tools drop-down menu.
2. Select Options.
3. Select Users from the Options submenu.

The Manage User Accounts dialog box appears.



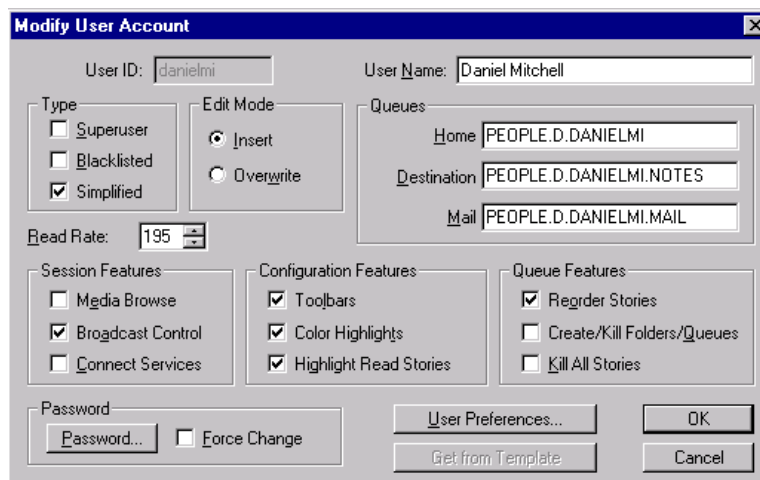
An asterisk (\*) in the User ID field will result in all user accounts listed when you click Search or press Enter.

4. Enter the user name in the User ID field.
5. Click Search or press Enter. The results of the search appear in the User List field located in the center of the dialog box.
6. Do one of the following:
  - a. Double-click the user name in the User List field.
  - OR-
  - b. Click the name once to select it, and then click the Modify/Display button.



The **Modify** button will appear with the word **Display** on it if you do not have authority to modify user accounts. This applies to user managers (umanager) who cannot alter superuser accounts. Also, the traits shown in the dialog box will appear gray to indicate that the information is for viewing only.

The Modify User Account dialog box appears.



The dialog box shows user traits associated with the account you chose, such as the user's name, read rate, and mail queue name. All the traits shown in the various sections of the Modify User Account dialog box are explained in detail in "Modify User Account Dialog Box" on page 4-4.

## Modifying User Traits

You must be logged on as a superuser or user manager (umanager) to change user traits. For an explanation of the umanager account and privileges, see "Creating a User Manager Account" on page 4-35.

To modify a user's traits from an Avstar Workstation, do the following:

1. Access the Modify User Account dialog box as explained in "Viewing User Accounts" on page 4-2.

2. Select or deselect check boxes, as required. Fill in the fields in the Queues section of the dialog box. See “User Traits Summary” on page 4-5 for more information.

You can click the Get from Template button to copy traits from another pre-defined user account. The template must be selected prior to the start of account modification or the button will be inaccessible (grayed out). See “Copying User Traits to Another User Account” on page 4-26 for more information.

3. Change or setup a password, as explained in “Changing a User’s Password” on page 4-9.
4. Click User Preferences and modify settings, as explained in “Changing User Preferences” on page 4-10.
5. Click OK to save modifications, or click Cancel to close the dialog box without saving changes.

## Modify User Account Dialog Box

**Modify User Account**

User ID:  User Name:

Type:  Superuser  Blacklisted  Simplified

Edit Mode:  Insert  Overwrite

Queues: Home   
Destination   
Mail

Read Rate:

Session Features:  Media Browse  Broadcast Control  Connect Services

Configuration Features:  Toolbars  Color Highlights  Highlight Read Stories

Queue Features:  Reorder Stories  Create/Kill Folders/Queues  Kill All Stories

Password:   Force Change



The Modify User Account dialog box divides the user's traits into sections, such as Type, Edit Mode, Queues, and so forth. These sections are explained in the following summary of all user traits.

## User Traits Summary

### User Name

The User Name field contains the user's real name. It should not be confused with the User ID, which the system uses to identify account activity. For instance, Daniel Mitchell may have an account with a user ID `danielmi`; his real name is Daniel Mitchell, but he will type `danielmi` to log on to Avstar NRCS.

### Type

The Type section contains the check boxes that determine what type of user account is assigned to the user, and consequently, what privileges. If the check box is selected, the type is applied to that user account.

- **Superuser** – A superuser account allows the user complete access to administration features, such as user accounts, the database, the System directory, and connect sessions to the Avstar console that controls the servers.
- **Black Listed** – A black listed account cannot be used to log in to an Avstar Workstation. This type is used for special accounts, such as `umanager` and `dbmanager`. It is not intended for standard user accounts.
- **Simplified** – A simplified account sets certain access limits, such as the maximum number of Avstar Workspaces allowed. See “Simplified User Settings” on page 4-19 for more information.

## Edit Mode

The Edit Mode section's radio buttons set up the condition of the PC keyboard's Insert key at log in.

- **Insert** – The Insert editing mode, when selected, means if a user types text between two characters, the text is inserted at the cursor position without overwriting the character to the right of the cursor.
- **Overwrite** – The Overwrite editing mode, when selected, means if a user types text between two characters, the character to the right of the cursor is replaced with the new text.

## Queues

Avstar NRCS provides a People directory in the database file structure that allows the system administrator to set up a personal directory and two queues for each user as data storage. The Queues fields in the Modify User Account dialog box indicate the navigation paths (or locations) of the user's personal directory and queues.

The image shows a dialog box titled "Queues" with three text input fields. The first field is labeled "Home" and contains the text "PEOPLE.D.DANIELMI". The second field is labeled "Destination" and contains the text "PEOPLE.D.DANIELMI.NOTES". The third field is labeled "Mail" and contains the text "PEOPLE.D.DANIELMI.MAIL".



*The actual directory and queues are not created here. See “Creating a New User Area in the News Database” on page 4-23 for more information.*

- **Home** – The Home field contains the path to the directory (folder) where the Destination and Mail queues are stored in the database file structure.

- **Destination** – The Destination field contains the path to the queue provided for the user as a storage location, such as Notes.
- **Mail** – The Mail field contains the path to the user's Mail queue, which is where all e-mail to that user is kept in the database.

### Read Rate

The Read Rate is the user's spoken reading rate in words per minute. The average English reading rate is 180 words per minute. Avstar NRCS uses the read rate of the designated user (presenter) to determine the audio (air) time of a story.

### Session/Configuration/Queue Features

There are three sections of the Modify User Account dialog box pertaining to features.

Session Features	Configuration Features	Queue Features
<input type="checkbox"/> Media Browse	<input checked="" type="checkbox"/> Toolbars	<input checked="" type="checkbox"/> Reorder Stories
<input checked="" type="checkbox"/> Broadcast Control	<input checked="" type="checkbox"/> Color Highlights	<input type="checkbox"/> Create/Kill Folders/Queues
<input type="checkbox"/> Connect Services	<input checked="" type="checkbox"/> Highlight Read Stories	<input type="checkbox"/> Kill All Stories

The Session Features section defines access to other parts of the system, such as the Video server.

- **Media Browse** – The Media Browse check box determines access to search the Video server.
- **Broadcast Control** – The Broadcast Control check box determines access to the Avstar Broadcast Control System (BCS) workstation—typically operated by the technical director.
- **Connect Services** – The Connect Services check box determines access to any services defined in the system.

The Configuration Features section pertains to the look of the Avstar Workspace.

- **Toolbars** – The Toolbars check box determines whether the user can create custom toolbars.
- **Color Highlights** – The Color Highlights check box determines whether the user can customize the highlighting status colors in the queue.
- **Highlight Read ...** – The Highlight Read Stories check box specifies that unread stories in the queue are highlighted on the user's screen. The highlight is removed when the cursor is positioned on the story.

The Queue Features section pertains to access privileges in the Queue panel of the Avstar Workspace.

- **Reorder Stories** – The Reorder Stories check box determines authority to alter the order of the stories in a queue.
- **Create/Kill ...** – The Create/Kill Folders/Queues check box determines authority to create or delete queues and folders (directories) in the database file structure, as seen in the Directory panel of the Avstar Workspace.
- **Kill All Stories** – The Kill All Stories check box determines authority to delete all stories from a queue at one time. The data is actually moved from the selected queue to the DEAD queue where it remains (and can be accessed) until purged.

### Password

- **Password** – The Password button opens a dialog box that you can use to set up or change the password protecting access to the user account.

See “Changing a User's Password” on page 4-9 for more information.

See “Forcing Individual Users to Change Their Passwords” on page 18-6 for more information.

- **Force Change** – The Force Change check box determines whether the user is forced to change the assigned password the next time he logs on.

### User Preferences...

The User Preferences button is used to view and/or modify a user’s preferences, such as keyboard, printer, and confirmation settings. See “Changing User Preferences” on page 4-10 for more information.

### Get from Template...

The Get from Template button is only used when copying the traits of one user’s account to another. See “Adding a New User Account” on page 4-26 for more information.

## Changing a User’s Password

To change a user’s password, do the following:

1. Click Password in the Modify User Account dialog box. The Change User’s Password dialog box appears:

2. Type the password in the New password field.
3. Confirm the new password by retyping it in the Confirm new password field.



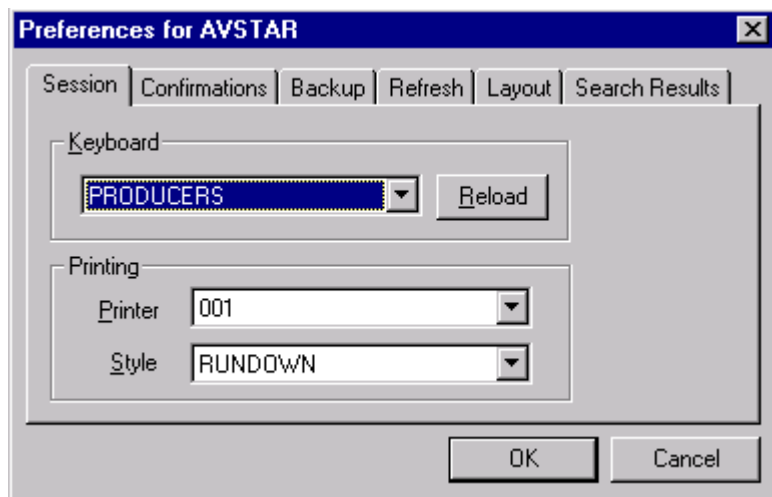
*The password must be a minimum of five alphanumeric characters (and a maximum of 12 characters) with no spaces. Use the system profile to set or change a required length for all passwords for your site.*

4. Do one of the following:
  - a. Click OK to save the password and close the dialog box.
  - OR-
  - b. Click Cancel to close the dialog box without saving changes.

## Changing User Preferences

To change user preferences, do the following:

1. Click User Preferences in the Modify User Account dialog box. The Preferences dialog box will appear, containing several tabs.



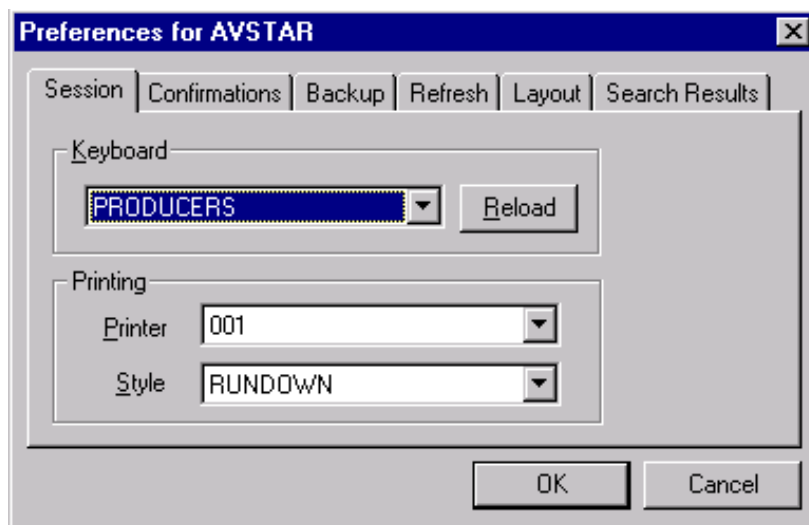
2. Modify the preference settings on each tab as needed. The settings are described in detail in the next section of this chapter.
3. Do one of the following:
  - a. Click OK to save them and close the dialog box.
  - OR-
  - b. Click Cancel to close the dialog box without saving preference changes.

## Preferences Dialog Box

System administrators can use the Preferences dialog box to set up default preferences for users, such as a default printer. However, the Preferences dialog box is—by default—accessible to users, so they can alter these settings at any time. To learn how you, as the system administrator, can limit access to this dialog box and its features, see “Simplified User Settings” on page 4-19.

### Session Tab

The Session tab has two sections, which system administrators can use to set up default user preferences.



- **Keyboard**
  - The keyboard drop-down list contains a list of keyboards (or sets of macros) that can be assigned to the user account as a default for when he logs in. The Reload button allows the keyboard assignment to take effect without having the user log off and back on.
- **Printing**
  - The Printing drop-down lists contain two lists: one of Printers and another of

pre-defined Styles that can be assigned to the user account as defaults for when he prints data from an Avstar Workstation.


### Confirmations Tab

The Confirmations tab is divided into sections and contains check boxes that determine whether Avstar NRCS prompts the user to confirm a request before completing the command.



- **Drag/Drop section**
  - **Story Operations** – Avstar Workstation (ASWS), when Story Operations is checked, will display a confirmation message before moving a story when you use the mouse to drag it to its new position.
  - **Queue Operations** – ASWS, when Queue Operations is checked, will display a confirmation message before moving all stories in a queue when you use the mouse to drag them to their new position.

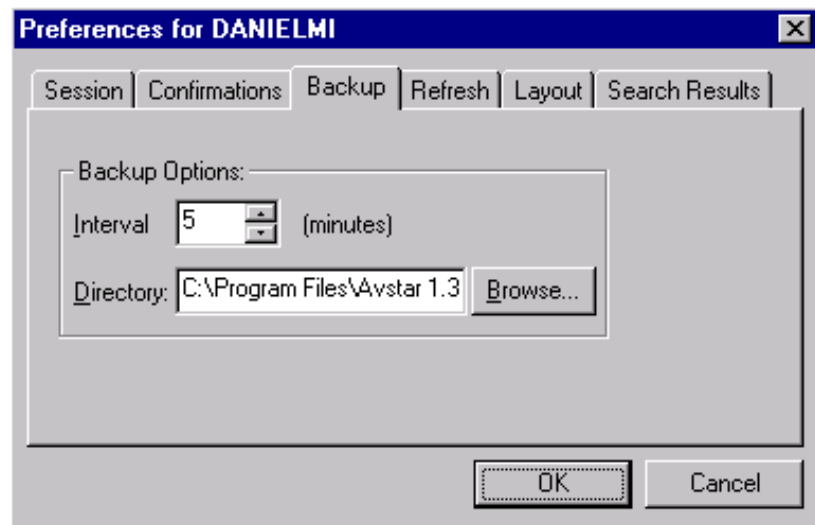


A production cue marker (shown  here) is called a groch or grommet and appears in the Story Text panel.

- **Queue Reorder** – ASWS, when Queue Reorder is checked, will display a confirmation message before moving a story to a new location in the same queue.
- **Delete/Kill section**
  - **Story** – ASWS, when Story is checked, will display a confirmation message before deleting a story or stories.
  - **Mail or Message** – ASWS, when Mail or Message is checked, will display a confirmation message before deleting e-mail or instant messages.
  - **Production Cue** – ASWS, when Production Cue is checked, will display a confirmation message before deleting a production cue and its marker from a story.
- **Saving Story** – ASWS, when Saving Story is checked, will display a confirmation message to save changes before closing an edited story. The confirmation also appears as a user moves the cursor from one story to another when building a show rundown. If you do not select the Saving Story check box, the system automatically saves changes before closing stories.
- **Large Print Jobs** – ASWS, when Large Print Jobs is checked, will display a confirmation message before sending potentially large print jobs to the printer, such as the text of all stories in a queue.
- **Exit** – ASWS, when Exit is checked, will display a confirmation message when the user attempts to close the Avstar NRCS program at the workstation.

## Backup Tab

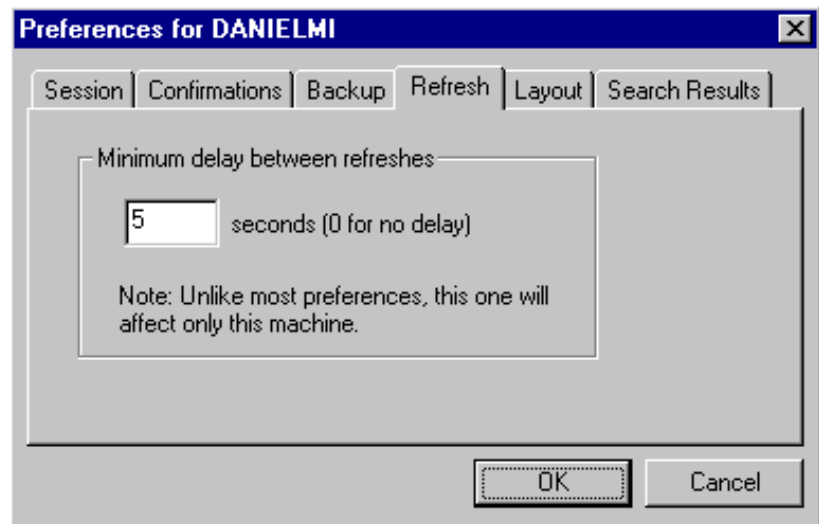
The Backup tab defines the settings for the Avstar Workstation to automatically back up work in a current session to a specified location at specific time intervals.



- **Interval** – Interval specifies the number of minutes between story backups. The default is 10 minutes. Set the interval to 0 (zero) minutes to turn off the automatic backup feature.
- **Directory** – Directory specifies the path name—the location in which ASWS should store backup copies of stories. The location should be a directory (folder) on the harddrive of the local PC/workstation. You can type the path in manually, or click the Browse button to select the directory from the Browse dialog box.

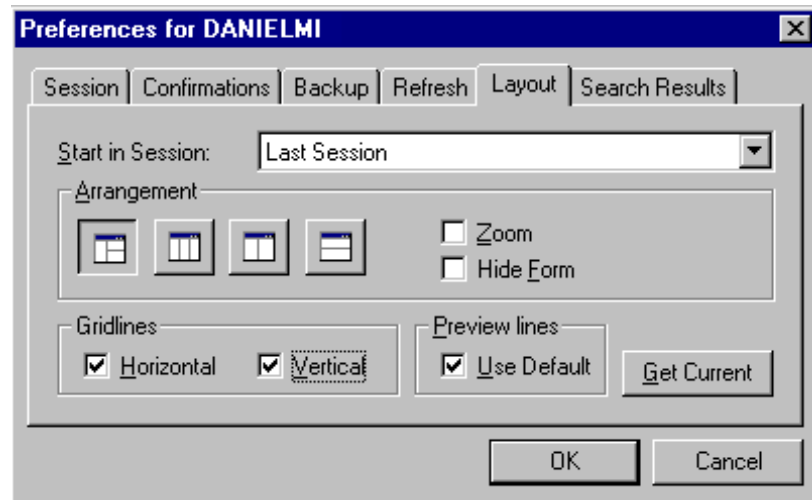
## Refresh Tab


The Refresh tab sets the seconds for refreshing the screen at the workstation. This preference is unique because it only affects the workstation on which it is set. Set the number to zero (0) for instant updating—that is, if you do not want to delay refreshes.



## Layout Tab

The Layout tab is divided into sections and contains buttons and check boxes that determine the layout of panels and workspaces in the Avstar Workstation main window.



- **Start in Session** – The Start in Session field specifies the default session that will appear on screen when the user logs in to Avstar Workstation (ASWS).
- **Arrangement section**
  -  – ASWS, depending on which Arrangement button is selected, will display the panels of the Avstar Workspace accordingly.
  - **Zoom** – ASWS, when Zoom is checked, will display the Avstar Workspace in zoom mode—that is, zoomed into one of the three panels: Directory, Queue or Story.

- **Hide Form**
  - ASWS, when Hide Form is checked, will display the Story panel with its Story Form panel hidden by default. The user can choose to show the Story Form panel, by selecting the option to show the form from the Story drop-down menu.
- **Gridlines section**
  - **Horizontal**
    - ASWS, when Horizontal is checked, will display horizontal gridlines between rows in the Queue panel.
  - **Vertical**
    - ASWS, when Vertical is checked, will display vertical gridlines between columns in the Queue panel.
- **Get Current**
  - The Get Current will reset the preferences on the Layout tab to what was set when the dialog box opened.
- **Preview Lines**
  - **Use Default**
    - ASWS, when Use Default is checked, will display the default number of preview lines for each story in the Queue panel as defined by the queue's properties. When Use Default is not checked, every queue will display only one line of information per story in the Queue panel by default.



*Users can manually override the Preview Lines setting by selecting the Story Preview option in the View drop-down menu. In the Story Preview dialog box, the user can override the default setting by typing in a number in the Lines to preview field. This overrides the setting for the queue while displayed. Once the user exits the queue, the queue's default setting is reinstated. If the user wants to return to the default setting manually (as defined in the queue's properties), the user can click the Default button in the Story Preview dialog box.*

The following figures show the difference between Queue panel displays with and without Preview lines. The first is the display using the default of seven preview lines, as defined in the queue's properties.

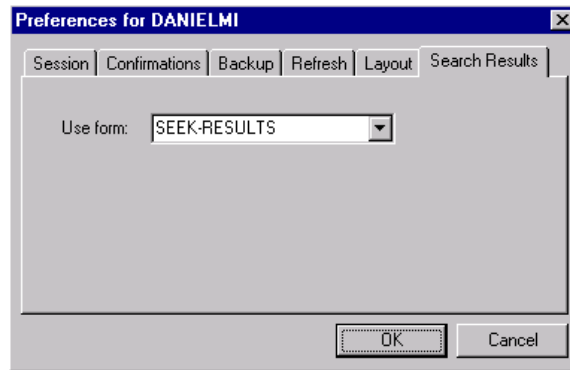
Slug	Lines	From	Mov
AM-IA--NewsBriefs 1stAdd 08-30	47	APk0ia--nr	8/30,
AM-IA--NewsBriefs 1stAdd 08-30 0380 AM-IA--News Briefs, 1st Add,370 Eds: Moved for PMS siodubdeskaccar12 SIOUX CITY, Iowa (AP) -- Motorists, including some police officers in their squad cars, were stranded Sunday night when a sudden rainstorm hit the Sioux City area. Hard-hit areas included sections of U.S. Highway 75 northbound and some entrance and exit ramps to Interstate 29.			
AM-BRF--LightsRecall 08-30 020	26	APp-----wr	8/30,
AM-BRF--LightsRecall 08-30 0205 AM-BRF--Lights Recall,0204 More than 17,000 work lights recalled WASHINGTON (AP) -- Around 17,300 work lights were recalled Monday because of fire, shock and electrocution hazards. The Consumer Product Safety Commission recalled more than 16,000 florescent work lights distributed by Wilmar Corp. of Kent, Wash. The lights have wiring that cannot carry adequate wattage. In addition, about 1,300 work lights made by Foremost Tool Co., also of Kent, Wash., have undersized			
AM-PneumoniaTest 08-30 0493	56	APp-----wr	8/30,
AM-Pneumonia Test,0493 FDA approves test to help diagnose pneumonia in 15 minutes By LAURAN NEERGAARD AP Medical Writer WASHINGTON (AP) -- Doctors struggling to decide whether a wheezing patient needs antibiotics to treat pneumonia are about to get a simple, 15-minute test to help their diagnosis. The Food and Drug Administration approved Binax, Inc.'s first-of-a-kind pneumonia test Monday. It uses			
AM-IA--MarcianoCrash Bjt 08-30	51	APk0ia--nr	8/30,

The second figure shows the Queue panel display without preview lines. This is the system's default view, unless otherwise specified.

Slug	Lines	From	Mov
PM-InternetHoax 08-30 0482	54	APp-----ar	8/30,
AM-RUN--Michael'sMagic Bjt 08-	89	APt-----sr	8/30,
AP-WI--ARX-FireWeatherFo 08-30	97	APv2wi-ooa	8/30,
AP-15thNewsAgenda 08-30 1044	126	APv1ntg-dr	8/30,
AP-HurricaneDennisInterm 08-30	71	APv2nwshaor	8/30,
AP-PMPrep-ScoresandSkeds 08-30	37	APv1al-cr	8/30,
AP-HurricaneDennisInterm 08-30	69	APv2nwshaor	8/30,
AP-PMPrep-TV/Birthdays 08-30 0	51	APv1al-cr	8/30,
AX-ABCN1:BabyGiraffe 08-30 017	35	APvabc-xtr	8/30,
AX-ABCN1:DrugSurvey 08-30 0441	56	APvabc-ntr	8/30,
AX-ABCN1:MoreNCDennis 08-30 01	42	APvabc-ktr	8/30,
AM-IA--Bullet-proofVests 08-30	41	APk0ia--nr	8/30,
AX-ABCN1:StateDept 08-30 0401	55	APvabc-ktr	8/30,
AM-IA--DryHydrants Bjt 08-30 0	57	APk0ia--nr	8/30,
AM-Lebanon-Israel 08-30 0267	32	APp-----ir	8/30,
AM-IA--NewsBriefs 1stAdd 08-30	47	APk0ia--nr	8/30,
AM-BRF--LightsRecall 08-30 020	26	APp-----wr	8/30,
AM-PneumoniaTest 08-30 0493	56	APp-----wr	8/30,
AM-IA--MarcianoCrash Bjt 08-30	51	APk0ia--nr	8/30,
AP-15thNewsMinute 08-30 0309	39	APv1al-hr	8/30,
AP-PMPrep-Segue 08-30 1112	145	APv1al-cr	8/30,
AX-ABCN1:SETopStory 08-30 0269	46	APvabc-vtu	8/30,

## Search Results Tab

The Search Results tab allows you to set the default form used in the Queue panel of the Search Results workspace.



- **Use form:**
  - ASWS will use the form selected from the Use form: drop-down list when displaying the results from searches in Avstar NRCS. The forms you can choose from are those created and stored in `SYSTEM.FORMS`. See “Creating a Form” on page 8-2 for more information.

## Simplified User Settings

A simplified user is one that has certain limitations pertaining to the Avstar Workstation. As the system administrator, you can define the limitations and then assign them to users. Only one set of limitations can be defined, which is then applied to all user accounts with the simplified user trait. In other words, either a user account has the simplified user trait, with its designated limitation settings, or it does not.

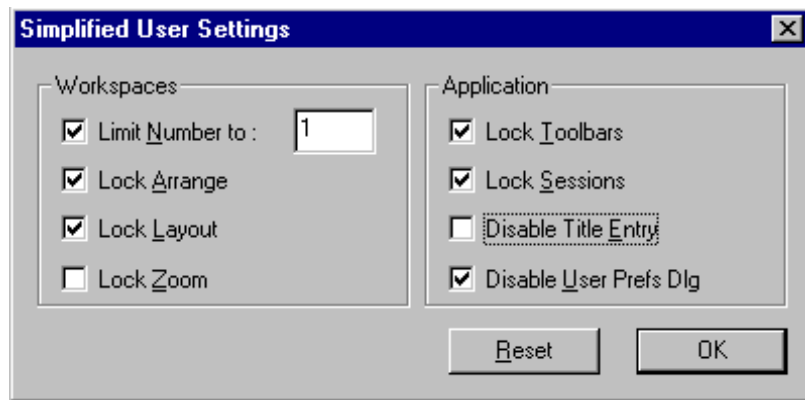
Some of the Simplified User Settings lock the user’s preferences to those defined by the system administrator using the Preferences dia-

log box. See “Changing User Preferences” on page 4-10 for more information.

To set up or modify the simplified user limitations, do the following:

1. Open the Manage User Account dialog box as explained in “Viewing User Accounts” on page 4-2.
2. Click the Simplified UI button.

The Simplified User Settings dialog box appears.



3. Select or deselect check boxes, as required.
4. Do one of the following:
  - a. Click the Reset button, to discard changes and reset the check box settings to what they were when the dialog box opened.-OR-
  - b. Click the OK button to save the settings and close the dialog box.

## Simplified User Setting Dialog Box

The Simplified User Settings dialog box divides the limitation check boxes into two sections: Workspaces and Application.



## Workspaces

- **Limit Number to:** – Avstar Workstation (ASWS), when Limit Number to: is checked, will prevent the user from opening more workspaces than the number specified. This numerical limit does not apply to the workspaces opened using the Urgent Wire and Mail buttons. However, this does lock the Urgent Wire workspace so the user is unable to navigate to other queues or directories in that workspace.
- **Lock Arrange** – ASWS, when Lock Arrange is checked, will prevent the user from altering the arrangement of panels in the Workspace. This disables the Arrangement buttons on the standard Layout toolbar. The setting is locked into the default arrangement as defined in the user's User Account Preferences.
- **Lock Layout** – ASWS, when Lock Layout is checked, will prevent the user from altering the layout of workspaces in the Avstar Workstation main window. This disables the Layout buttons on the standard Layout toolbar. The setting is locked into the default layout as defined in the user's User Account Preferences.
- **Lock Zoom** – ASWS, when Lock Zoom is checked, will prevent the user from altering the zoom of panels in the Avstar Workspace. The setting is locked into the default as defined in the user's User Account Preferences.

## Application

- **Lock Toolbars** – ASWS, when Lock Toolbars is checked, will prevent the user from altering the display of toolbars.

- **Lock Sessions** – ASWS, when Lock Sessions is checked, will prevent the user from creating or altering sessions. The user will be locked to sessions created prior to the Lock Sessions being applied to the user account.
- **Disable Title Entry**– ASWS, when Disable Title Entry is checked, will prevent user access to the Title Entry dialog box, used to enter production cues in stories, and the Edit Title Entry Template dialog box, used to create CG templates for the Title Entry feature.
- **Disable User Prefs...**– ASWS, when Disable User Prefs Dlg is checked, will prevent user access to the Preferences dialog box. The user will be unable to alter user preferences, such as default printer settings. The user will be locked to settings already in place at the time Disable User Prefs DLG is applied to the user account. See “Changing User Preferences” on page 4-10 and “Simplified User Settings” on page 4-19 for more information.

## Setting up New Users in Avstar

To set up new users in Avstar NRCS, you need to follow these three procedures, which are explained in detail in this section:

1. Create areas in the Avstar NRCS database file structure where the user can store notes and receive mail. See “Creating a New User Area in the News Database” on page 4-23 for more information.
2. Add a new user account so that your system recognizes the user. This includes setting up the user traits associated with the account. See “Adding a New User Account” on page 4-26 for more information.

3. Enable the new user to receive mail by adding him or her to the appropriate group. (See “Enabling a New User to Receive Mail” in this chapter.)

## Creating a New User Area in the News Database

Each user needs an area in the database to keep notes and to receive e-mail. Usually, these areas are separate queues called Notes and Mail. These queues are kept in a sub-folder—with the user’s account name—in the People directory.

The common practice is to separate the first level of People sub-folders by using the first initial of the user’s last name—otherwise, since the system is limited to 255 user names in the People directory, your site may eventually reach the limit.

For instance, the Home directory for user DANIELMI would be: PEOPLE . D . DANIELMI. The Notes and Mail queues would be: PEOPLE . D . DANIELMI . NOTES and PEOPLE . D . DANIELMI . MAIL, respectively.

You must be logged on to Avstar NRCS with a user account that has authority to create new directories and/or queues to complete this procedure.

To create a new directory (folder):

1. Using the database file structure in the Directory panel of the Avstar Workspace, select the directory under which you want the new folder to be created, as shown in the following example.



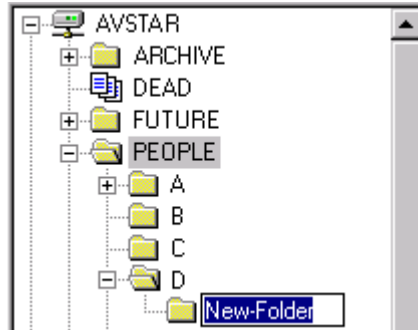
Select the PEOPLE folder, then the folder with the alphabetic name corresponding to the first letter of the user’s name, such as D. The new folder will be created in the D directory (folder), such as DANIELMI.

2. Do one of the following:
  - a. Click on the Tools drop-down menu, then select New Folder.

-OR-

- b. Right-click on the folder in the Directory panel, and choose New Folder from the pop-up menu.

A new folder is created under the selected folder. The New-Folder appears at the end of the list of existing folders. The title, New-Folder, is highlighted, so you can rename it.



3. Type the name of the new folder.
4. Press Enter to save the new folder name. You can now open the new folder (directory) by double-clicking on it.

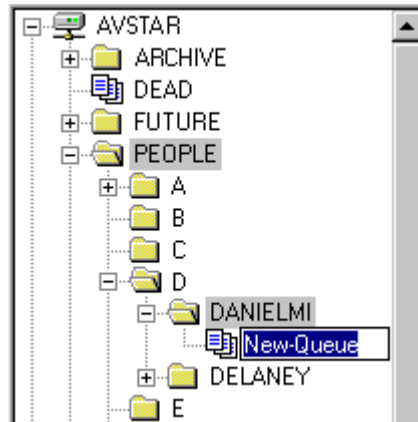
To create a new queue, such as the Notes and Mail queues for user DANIELMI, do the following:

1. Navigate to and select the folder created to hold the queue you want to create, such as PEOPLE . D . DANIELMI.
2. Do one of the following:
  - a. Click on the Tools drop-down menu, then select New Queue.

-OR-

  - b. Right-click on the folder in the Directory panel, and choose New Queue from the pop-up menu.

A new queue appears under the folder you selected. The New-Queue appears at the end of the list of existing queues. The title, New-Queue, is highlighted, so you can rename it.



3. Type the name of the new queue, such as **MAIL** or **NOTES**.
4. Press Enter to save the new queue name. You can now open the new queue by double-clicking on it.



*There are some restrictions pertaining to adding directories and queues in the Avstar database. See “Adding a Directory or Queue” on page 5-3 for more information.*

## Adding a New User Account

When adding a new user account, you have the option of creating the account from scratch, or copying the pre-defined traits of another user account already in the system. Both options are covered in this section.

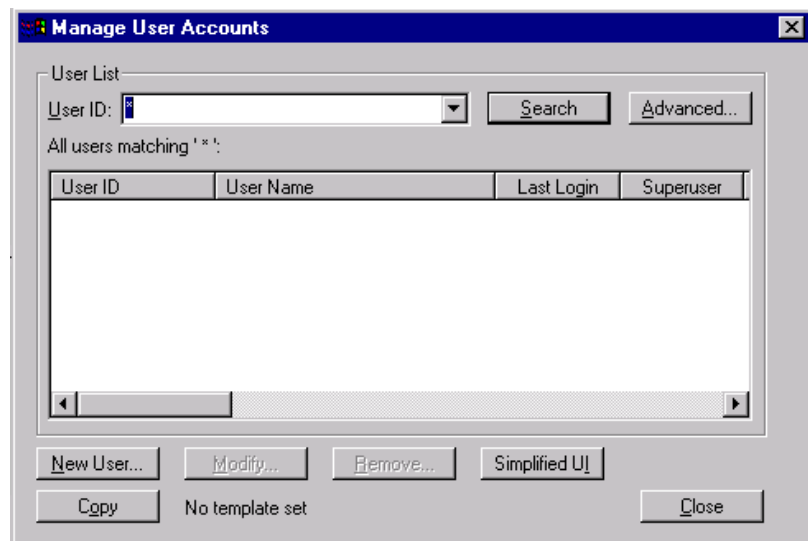
### Copying User Traits to Another User Account

Before you can copy user traits from one user account to another, you must first select the account you want to copy—that is, select an account to use as a template for the new account you are going to create.

To define an account as a template for copying to other accounts, do the following:

1. Click the Tools drop-down menu.
2. Select Options.
3. Select Users.

The Manage User Accounts dialog box appears.





*If you do not have superuser privileges, which permits access to the Manage User Accounts dialog box, the system will prompt you for the `umanager` password. If the `umanager` account does not exist in the system, then access is only allowed to system administrators—that is, those with superuser privileges. See “Creating a User Manager Account” on page 4-35 for more information. Also see “User Traits Summary” on page 4-5 for more information on the superuser trait and its privileges.*

4. Type in the User ID of the user account you want to copy—that is, use as a template. You can use the Search button to locate the user account if you do not know the User ID.
5. Click Copy. The User ID should appear to the right of the button. When no template is selected for copying, the words, “No template set,” appears to the right of the Copy button.

You are now ready to create a new user account and copy the user traits, or copy the traits to an account that already exists in the system.

## Creating a New User Account

To add a new user account:

1. Click the Tools drop-down menu.
2. Select Options.
3. Select Users.

The Manage User Accounts dialog box appears.

4. Click New User.

The Add New User dialog box appears.

5. In the User ID field, enter the login name of the user account.
6. (Optional) In the User Name field, enter the user's real name.
7. Do one of the following:
  - a. Click the appropriate check boxes for the user traits you want to set. See "User Traits Summary" on page 4-5 for more information.
  - OR-
  - b. Click the Get from Template button to copy traits for another pre-define user account. See "Copying User Traits to Another User Account" on page 4-26 for more information.
8. Click Add to add the new user account, or click Cancel to cancel the addition of the new user account.



## Enabling a New User to Receive Mail

This section gives you the basic steps you need to follow to enable a new user to receive mail. If you need more information, refer to Chapter 6, “Groups”.

To enable a new user to receive mail, do one of the following:

1. Add the user to a group in `SYSTEM.GROUPS`. When the group story is saved, the mail delivery files are updated automatically.

A group story is one that you created in the system for groups in your organization such as newscasters, staff, or reporters. By adding the user to a group, the user inherits the security traits that were set up for that group.

2. Run the `grpcheck` command on the Avstar Console to rebuild and update the mail alias file. If this file is not rebuilt, the user will be unable to receive mail. Type:

```
grpcheck system.groups
```

## Searching for Information About Users

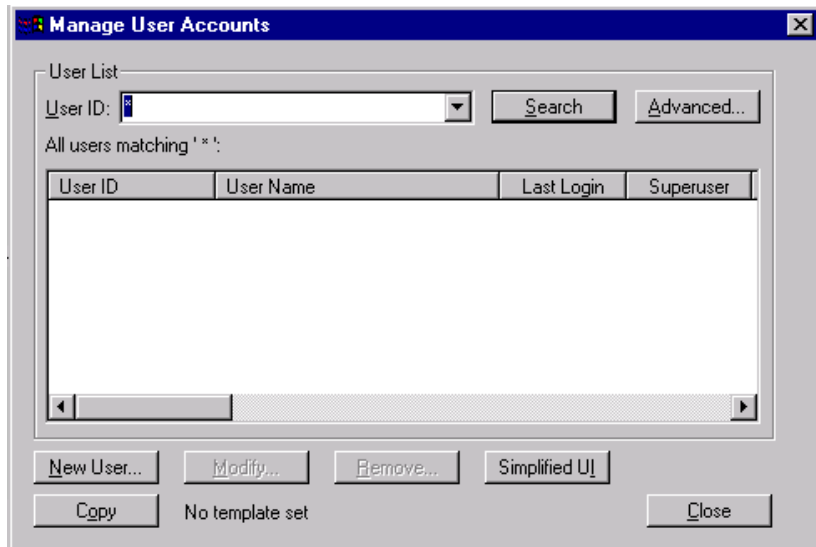
A search capability in Avstar NRCS enables you to search for information about a particular user by specifying a user name and including certain criteria to refine the search. You can specify any alphanumeric characters in the search. You can use the asterisk (\*), which acts as a *wildcard*, only as a suffix—not as a prefix or in the middle of a word. Used alone, the wildcard is equivalent to “all.” Used with additional information, the wildcard serves as a parameter to the search.

For instance, if you are searching for all user accounts beginning with Dave, type `Dave*` (no space).

To search for information about users:

1. Click the Tools drop-down menu.
2. Select Options.

3. Select Users. The Manage User Accounts dialog box appears.



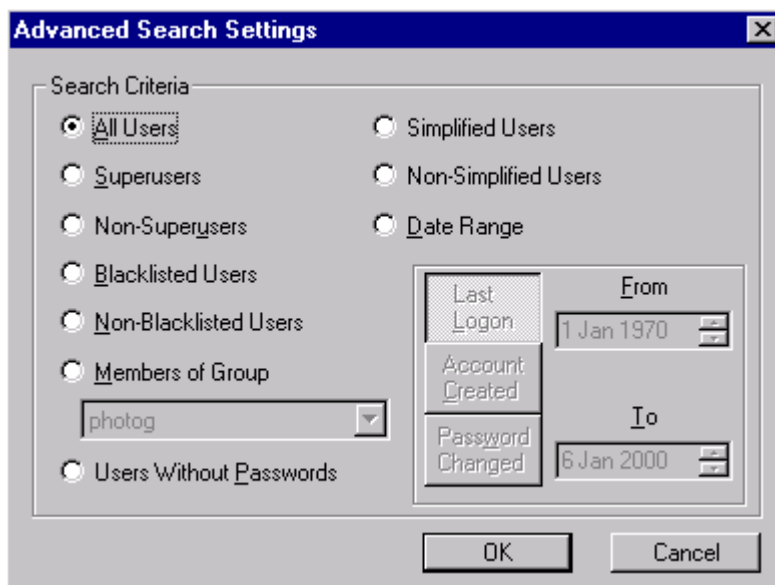
4. Type the name of the user in the User ID field and click Search.

If you search with a wildcard character and the system finds multiple matches, a results box appears listing all “hits”. You can specify one by double-clicking on it.

The results of the search appear in the User List field in the center of the dialog box.

To quickly locate a name in the User List, type the name you want. User names are not case-sensitive, so you can use lowercase. To prevent you from having to type the whole name, the system automatically tries to match the letters you supply with a name in the list. Continue typing until the system locates the name you want.

- Click Advanced to refine your search for a user, or to obtain more extensive user information. The Advanced Search Settings dialog box appears with the All Users setting selected by default.



- Select from the settings to specify additional search criteria. The criteria options are explained in detail in Table 4-1.

**Table 4-1 Advanced Search Criteria**

Setting	Description/Choice
All Users	Search through all users listed in the server.
Superusers	Confine the search to users with superuser accounts.
Non-Superusers	Confine the search to users with non-superuser accounts.
Blacklisted Users	Confine the search to user accounts that cannot be used to log in.

Setting	Description/Choice
Non-Blacklisted Users	Confine the search to user accounts that can be used to log in.
Members of Group	Confine the search to users belonging to the security group you select from the drop-down list.
Users Without Password	Confine the search to users who do not have passwords.
Simplified Users	Confine the search to user accounts that have the simplified user trait.
Non-Simplified Users	Confine the search to user accounts that do not have the simplified user trait.
Date Range	<p>Confine the search to dates you specify in the From and To fields and the kind of date range:</p> <ul style="list-style-type: none"> <li>• When the user last logged in</li> <li>• When the user account was created</li> <li>• When the password changed</li> </ul> <p>Specify the date by either clicking the arrow buttons or typing the dates in ddmmmyyyy format. Indicate the day with two digits, the month with three letters, and the year with four digits.</p>

7. Click OK to confirm your advanced search setting or click Cancel to cancel it.
8. Click Search to initiate the search.

A progress bar appears if a lengthy search is underway.

The results of the search appear in the User List field in the center of the Manage User Accounts dialog box. Above the field, Avstar NRCS

will display a brief statement indicating what matched the search criteria, such as: All users matching '\*':

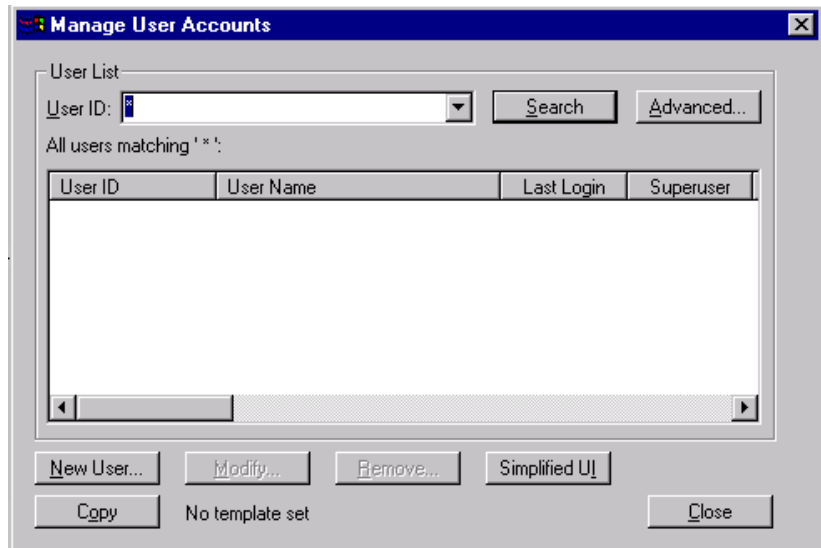
Use the horizontal scroll bar at the bottom of the User List field to view the information headings, such as User Name, Last Login, Read Rate, and so forth.

## Removing User Accounts

You must have access to the Manage User Accounts dialog box to remove user accounts. In other words, you must be logged on as a system administrator—that is, with an account that has superuser privileges—or user manager (`umanager`) to remove user accounts.

To remove user accounts, do the following:

1. Click the Tools drop-down menu.
2. Select Options.
3. Select Users. The Manage User Accounts dialog box appears.





*If you are logged on as a system administrator, the Manage User Accounts dialog box will appear automatically, following step 3. Otherwise, Avstar NRCS will prompt you for the `umanager` password (if that account exists in the system) before allowing access to the dialog box.*

4. Enter the user name in the User ID field.
5. Click Search or press Enter. The results of the search appear in the User List field.
6. Select the name of the user you want to remove by clicking the name in the User List field.
7. Click Remove.
8. Click OK to remove the user or Cancel to stop the removal.



*After removing the user, you will also need to remove the user's Home directory and the Notes and Mail queues by deleting them from the system's database file structure.*

## Creating a User Manager Account

A user manager has some special system privileges, but not as many as a system administrator/superuser. For instance, user managers can access the Avstar NRCS database and add or change any user account, except a superuser account.

There can be only one user manager account in Avstar NRCS; however, several users can have user manager privileges. Unlike a superuser account, the user manager account is not used to log in to the system. Users with user manager privileges log in to their own accounts, as usual. When they need to do user manager tasks, they must access the Manage User Accounts dialog box, by typing in the user manager password.

To create a user manager account, do the following:

1. Create a user account as explained in “Adding a New User Account” on page 4-26.
2. Give the account a User ID: `umanager`.
3. Assign a password to this account.
4. Make the `umanager` account blacklisted so that no one can use it to log in.
5. Assign the account superuser status to prevent a user manager (or anyone who is not a superuser) from changing the `umanager` password.
6. Tell the user managers the password for the `umanager` account.



**For further security, a write-access group should be assigned to `SYSTEM.GROUPS` and only those with user manager privileges should be included in the group. If no write-access group is assigned to `SYSTEM.GROUPS`, then any user who knows the `umanager` password can access the Manage User Accounts dialog box by clicking the Tools drop-down menu, selecting Options, then choosing Users. Once a write-access group is set up, any user managers with nonsuperuser accounts must be included in the write-access**

group for *SYSTEM.GROUPS* or they will not be allowed access to the Manage User Accounts dialog box. See “Groups Tab” on page 5-34, “Users as Members of a Group” on page 6-15, and “Restricting Both Reading and Writing” on page 6-26 for more information.

## Creating a Database Manager Account

A database manager has some special system privileges, but not as many as a system administrator/superuser. For instance, database managers can add or change any database trait to a directory or queue in the Avstar NRCS database from a workstation. Database managers also have access to the CG Template Editor, used to create and modify template for Avstar Title Entry feature.

There can be only one database manager account in Avstar NRCS; however, several users can have database manager privileges. Unlike a superuser account, the database manager account is not used to log in to the system. Users with database manager privileges log in to their own accounts, as usual. When they need to do database manager tasks, they must access the Directory/Queue Properties dialog box. To modify anything in the dialog box, they must provide the database manager password.

To create a database manager account, do the following:

1. Create a user account as explained in “Adding a New User Account” on page 4-26.
2. Give the account a User ID: `dbmanager`.
3. Assign a password to this account.
4. Make the `dbmanager` account blacklisted so that no one can use it to log in.
5. Assign the account superuser status to prevent a user manager (or anyone who is not a superuser) from changing the `dbmanager` password.



6. Tell the database managers the password for the `dbmanager` account.

## Logging out All Users

Sometimes maintenance of the Avstar system requires you to first log out all users before completing a certain task, such as shutting down the system. This section explains the best way to log out all users from the Avstar console.

To log out all users, do the following:

1. Select all servers at the console and type `offline` to take the system offline. The `offline` command prevents users from logging in.
2. Select all servers and type `broadcast` followed by the message warning users already logged in that they need to log out. Include when and why in the message, such as the time the system will be shut down. Here are a couple of examples:

```
AVSTAR-A: broadcast WARNING!System shut down 12PM
```

```
AVSTAR-A: broadcast WARNING!Log Out within 5 Min.
```

3. At the specified time, select one server and type the command `list s` to check who is still logged in. A message similar to the following appears:

T11	milller	A
T82	allen	B
T101	stevens	A
R801	stevens	A

The `list s` command lists: the device controlling the session, the user account used for the session, and the server servicing the session.

4. Do one of the following:
  - a. Send another broadcast instructing them to log out now.
  - OR-
  - b. Select all servers and type `logout all` to log out all users.



*If a user is editing a story, it is saved before the system logs out the user.*

5. Type `list s` again to check for connect session users.

The `logout all` console command does not log out users who are currently in a connect session.

```
AVSTAR-A: list s
```

```
T101  stevens      A
```

```
R801  stevens      A
```

If any users are still logged in, notify them of the shutdown by some other means, such as by telephone.



**If a user is in a connect session when you shut down the system, the user's workstation stops, the session is disconnected, and any unsaved work is lost. Ensure any connect session users have logged out before you shutdown the system.**

# CHAPTER 5

## *Stories, Queues, and Directories*

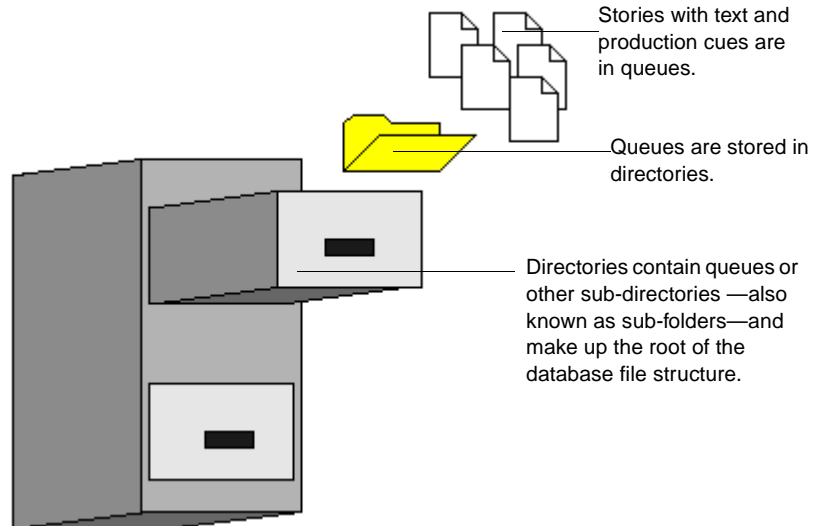
All relevant Avstar information—except the system software and the UNIX files—is stored in the Avstar NRCS database. This database contains scripts, rundowns, e-mail, messages, and any other kind of information that is entered into the system. Some database maintenance, such as altering the database file structure and traits, can be done from the Avstar console or from any Avstar Workstation. This chapter focuses on maintenance tasks at the workstation when possible. However, when a task can be done at both the workstation and console, the console information is provided as an appendix in this manual. See Appendix G, “Managing Traits at Console,” for more information.

This chapter contains the following sections:

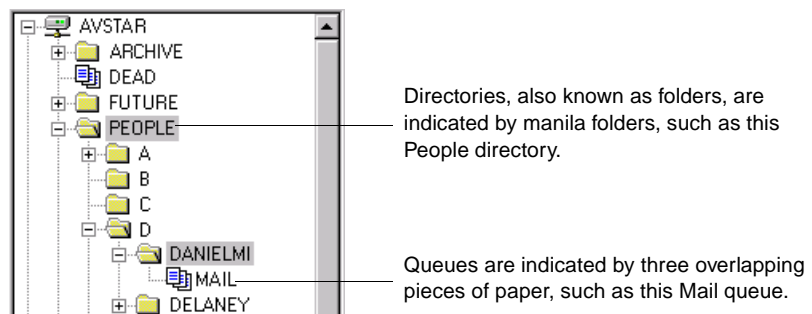
- Overview
- Adding a Directory or Queue
- Viewing Database Traits
- Changing Database Traits
- Database Traits Summary
- Locking and Unlocking

## Overview

The Avstar NRCS database is where all the data, such as scripts, run-downs, user accounts, and so forth, are stored. The database is structured in a way to promote ease of maintenance. For instance, it contains a file structure made up of directories, that contain other folders or queues, which in turn contain stories. It works similar to a filing cabinet.



In Avstar NRCS, the database file structure can be seen (depending on your access privileges) in the Directory panel of the Avstar Workspace.



The scripts, notes, e-mail, news stories, and other kinds of information are all called stories; each story is associated with a particular queue, and each queue with a single directory. A directory can also be associated with other directories.

Directories and queues have database traits that capture information about how the system manages the stories they contain, and also what actions users can perform with those stories.

For instance, by modifying the database traits of a particular queue, you can:

- Set its stories to read-only
- Restrict who can read them
- Enable the system to routinely purge stories in the queue that are older than a certain number of days

Database traits are used in conjunction with the user traits discussed in **Chapter 4, “Users”**. For instance, stories in a queue with a security restriction set as a database trait can be read only by users whose user traits include the appropriate security level.

## Adding a Directory or Queue

Before you can modify the database traits of queues or directories, the queues and directories must exist in the database. If they do not, you can create them from the Avstar Workstation.

### A Few Restrictions:

There are certain restrictions you should be aware of when creating new directories and queues.

- The total path name of a directory, including the separator characters (.), cannot exceed 60 characters.
- The total path name of a queue, including the separator characters (.), cannot exceed 62 characters.

- Each branch of a path name—that is the name between periods—cannot exceed 20 characters.
- The number of directory levels available is limited to 31.
- You cannot use the space, period, “at” symbol (@), backslash (\), or forward slash (/) characters in directory or queue names.
- The system has a limit of 255 queues per directory.



*The 255 limit also applies to first-level sub-folders in a directory. If you need more than 255 in a directory, such as personal employee folders, create alphabetic sub-folders on the first level, then place the personal folders in the matching sub-folder. For instance, in a given directory, you could have 26 sub-folders, each with one of the 26 letters of the English alphabet as a name. This enables you to have 255 personal folders in each of the 26 alphabetic sub-folders, or up to 6,660 employees. See “Creating a New User Area in the News Database” on page 4-23 for more information.*

Ordinarily, directories and queues are listed in alphabetical order within their parent directory. However, you can add items to a directory in a different order. For instance, if you had directories for each month in the Futures directory, you would want them to appear in order by month (January, February, and so on). To do this: turn on the sequential database trait for the parent directory before you create the new items. See “Database Traits Summary” on page 5-25 for more information.

## Creating a New Directory

To create a new directory (folder):

1. Log in as a superuser unless you have write-access to the parent queue or directory. This ensures that you have full access to the database.

2. Using the database file structure in the Directory panel of the Avstar Workspace, select the directory under which you want the new folder to be created, as shown in the following example.



For instance, you could select the Shows folder if you wanted to add a new directory for a 10PM show. Once the new sub-folder (10P) is created, you can create queues or additional sub-folders in it.



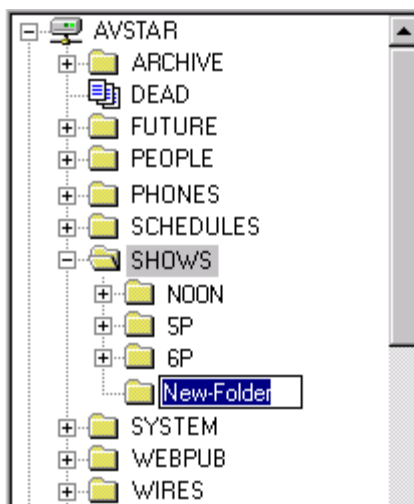
*If you are creating a new first-level directory, be sure to select the server rather than a directory (folder).*

3. Do one of the following:
  - a. Click on the Tools drop-down menu, then select New Folder.

-OR-

  - b. Right-click on the directory—or server, if you are creating a new first-level folder—in the Directory panel, and choose New Folder from the pop-up menu.

A new folder is created under the selected folder. The New-Folder appears at the end of the list of existing folders. The title, New-Folder, is highlighted, so you can rename it.

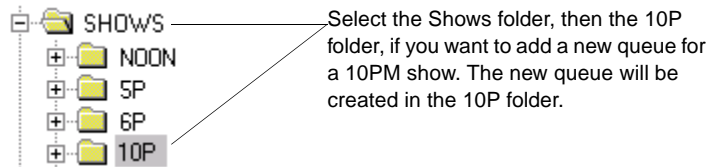


4. Type the name of the new folder, such as **10P**.
5. Press Enter to save the new folder name. The newly created folder will inherit the database traits of its parent directory initially. You can open the new folder by double-clicking on it.

## Creating a New Queue

To create a new queue, such as the Rundown and Master queues for the 10PM show, do the following:

1. Navigate to and select the directory (folder) created to hold the queue you want to create.



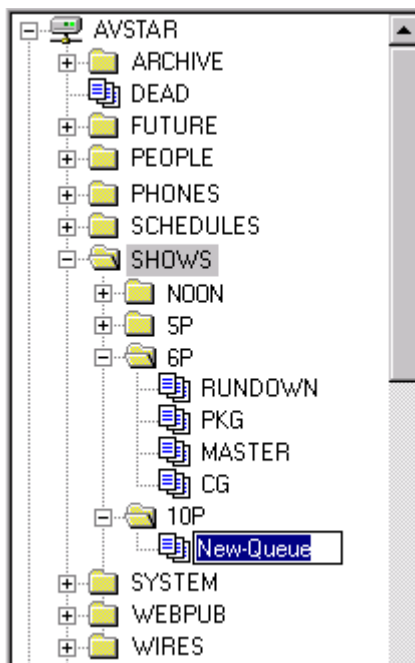
2. Do one of the following:
  - a. Click on the Tools drop-down menu, then select New Queue.

-OR-

  - b. Right-click on the folder in the Directory panel, and choose New Queue from the pop-up menu.



A new queue appears under the folder you selected. The New-Queue appears at the end of the list of existing queues. The title, New-Queue, is highlighted, so you can rename it.



3. Type the name of the new queue, such as **RUNDOWN** or **MASTER**.
4. Press Enter to save the new queue name. The newly created queue will inherit the database traits of its parent directory initially. You can open the new queue by double-clicking on it.

## Setting up the Outgoing Mail Queue

When someone sends e-mail, the first thing your system does is move the mail to the *outgoing mail queue*, usually called `SYSTEM.MAIL.OUT`.



*The outgoing mail queue's name is defined in `/site/dict/queues`. When your system was installed, the outgoing mail queue was defined in this dictionary as `SYSTEM.MAIL.OUT`. To change this name, modify the dictionary entry in `/site/dict/queues`.*

Mail always uses the form that is assigned to the outgoing mail queue. See “Mail Form” on page 8-29 for more information.

Once the mail arrives in that queue, a utility program known as the mail server processes and sends it to its intended destination. If the outgoing mail queue, `SYSTEM.MAIL.OUT`, does not exist in the System directory, the mail server cannot distribute e-mail. To create this queue, follow the instructions provided in “Creating a New Queue” on page 5-6, making these adjustments:

- The queue’s name should be “Out.”
- The queue should be located in the Mail folder, located in the System directory. If the Mail folder (directory) does not exist, create it. See “Creating a New Directory” on page 5-4 for more information.

Moreover, the Out queue’s read group must be set properly for the mail server to process mail correctly. See “Read Group” on page 6-23 for more information. Although not required, you can restrict read permission for the queue `SYSTEM.MAIL.OUT` to a group that has no users. Doing so does not interfere with anyone’s ability to send e-mail, but it prevents anyone (except superusers) from reading mail that is in `SYSTEM.MAIL.OUT` waiting to be processed. See “Restricting Both Reading and Writing” on page 6-26 for more information.

Also, for your system to notify the mail server when new mail arrives in `SYSTEM.MAIL.OUT`, that queue must have the same mailbox number assigned to it as the mail server. If the queue does not have a mailbox or has an incorrect one, your system has no way to notify the mail server when there is mail to process. See “Mailbox section” on page 5-39 for more information.



*The mailbox number is not solely related to e-mail. This database trait lets you link a queue to a server (utility program) so that the system notifies the server program when stories are added to or edited in the queue. See “Mailbox section” on page 5-39 and Chapter 14, “Servers,” for more information.*

## Setting up the Dead Letter Queue

Your system must also have a *dead letter queue*, usually called `SYSTEM.MAIL.ERROR`. This queue is a final destination for any e-mail

that your system is unable to deliver to the addressee or return to the sender.



*The dead letter queue's name is defined in `/site/dict/queues` as `SYSTEM.MAIL.ERROR`. To change the name of the queue, modify its dictionary entry in `/site/dict/queues`.*

If `SYSTEM.MAIL.ERROR` does not exist, any mail that the mail server cannot deliver or return to the sender is put in the Dead queue. To create this queue, follow the instructions provided in “Creating a New Queue” on page 5-6, making these adjustments:

- The queue's name should be “Error.”
- The queue should be located in the Mail folder, located in the System directory. If the Mail folder (directory) does not exist, create it. See “Creating a New Directory” on page 5-4 for more information.

Returned mail may contain sensitive information. Therefore, restrict read permission for `SYSTEM.MAIL.ERROR` to a group that has no members. Then, only superusers can read mail in the queue. See “Read Group” on page 6-23 for more information. Examine the queue occasionally to see whether any mail exists.

## Creating a New Story

To create a new story in a queue, do the following:

1. Navigate to the queue in which you want to create a story.
2. Open the queue by double-clicking on it.
3. Press the Insert key. A new story is inserted at the top of the queue.

## Removing a Directory or Queue

If a queue or story is locked, unlock it first before removing it from the database.

Also, each directory or queue should be empty of other directories, queues, and stories, before it is removed, but it is not required.



**If a directory contains sub-folders or queues when you attempt to remove it, Avstar NRCS will prompt you for confirmation. If you affirm the remove command, the directory and all its contents will be removed from the system. Caution should be taken so that sub-folders and sub-queues are not inadvertently removed.**

To remove a directory or queue from the database, do the following:

1. Log in as a system administrator unless you have write-access to the queue or directory.
  2. Select the directory or queue you want to remove.
  3. Do one of the following:
    - a. Click the Tools drop-down menu, then select Remove Folder (or Remove Queue).
- OR-
- b. Right-click on the directory or queue, then select Remove Folder (or Remove Queue) from the pop-up menu.

## Renaming a Directory or Queue

You cannot change the name of a directory or queue from an Avstar Workstation. However, you can rename one from the Avstar console. All traits are preserved when a folder or queue is renamed.



**Do not rename queues on an active database. Do not run directory or queue modification console commands (such as `dbvisit` or `dbtraits`) at the same time as the rename command.**

To rename a directory or queue in the database, do the following:

1. Become a console superuser.

2. Select *all* computers. See “Selecting One or More Servers” on page 2-5 for more information.
3. Type the **offline** command.
4. Broadcast a message to users to log out.
5. Logout all users on the system, before renaming a queue or directory. This ensures that no stories are open for editing and no devices are running while `rename` is executing.



**If users are not logged out, changes to stories may not be saved after the queue or directory is renamed. It is often most efficient to make several name changes at once. See “Logging out All Users” on page 4-37 for more information.**

6. Type **stop all**. This command stops running all servers, wire programs, and devices.
7. Select *one* computer.
8. Type the `rename` command to rename the folder or queue. Use the following format:

```
rename [-v|-r] <from> <to>
```

For instance, to rename your `People` folder to `PEOPLE.STAFF`, select one computer and type:

```
AVSTAR-B# rename people people.staff
```

A message similar to the following appears:

```
Do you really want to rename    PEOPLE
and all its sub-directories to PEOPLE.STAFF ?
56 records will be modified [y/n]:
```

To display a console message for each renamed folder and queue, include the `-v` (for verbose) option with the `rename` command, such as:

```
AVSTAR-B# rename -v people people.staff
```



If an attempt to rename a folder or queue was interrupted by a system crash, complete it by re-entering the same command with the `-r` option. Use this option *only* to resume an interrupted renaming—at any other time, its use will corrupt the database.

9. To continue with the renaming, type `y`.

```
56 records will be modified [y/n]: y
Adding new directories...
Updating directory names...
56 directories renamed
1 directories added
```

The system verifies that the queue or folder you specified exists, and it creates new folders necessary to complete your command (such as `STAFF` in this example). If you choose a pathname over 63 characters, the following message appears:

```
TO: name too long
```

10. A verification request appears:

```
Do you want to update the user file (MAIL, HOME,
DEST)? [y/n]:
```

The user file is where the names of users' mail queues, home folders, and automatic destinations are stored. If you answer `y`, any item affected by renaming is changed automatically. If you answer `n`, you must change them yourself. Typically, answer `y`.

```
Do you want to update the user file (MAIL, HOME,
DEST)? [y/n]: y
```

```
23 user records modified
```

11. Manually update any other references on the Avstar system to the renamed folders.



Update the references while the system is unavailable to users. Failure to update any references affected by renaming a folder or queue can cause problems with system operation.

These references can include:

- Command bar icons set up by Avstar users
  - Your system's service table
  - Dialogs
  - Keyboard description stories
  - Server or Rx/Tx link job lists
  - Wire distribution or keyword stories
  - Your system's queue dictionary (/site/dict/queues)
12. Select *all* servers and type **restart all** to restart all devices.  
You see `Hot-to-go` messages as each device starts.
  13. Type **online** to bring the system back online.  
This allows users to log in.
  14. Exit from superuser mode by pressing CTRL-D.

## Viewing Database Traits

You can get information about your Avstar database from both the Avstar console and workstations. Which one you use depends on what information you want.

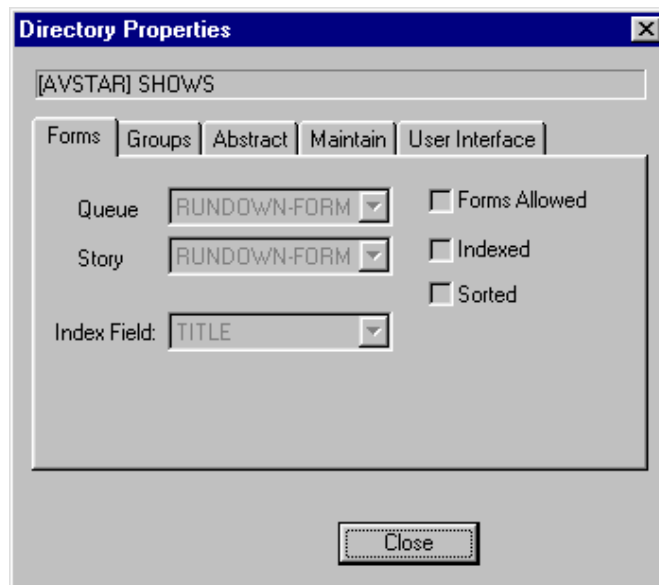
### From the Avstar Workstation...

To get database information on a specific directory or queue from an Avstar Workstation, do the following:

1. Log in to Avstar NRCS at a workstation.
2. Open an Avstar Workspace.
3. Move your mouse to the name of the directory or queue you want in the Directory panel.
4. Right-click on the directory or queue name.

5. Select Properties in the pop-up menu.

The Directory/Queue Properties dialog box shows you the properties (traits) for the directory or queue you selected; however, its look may vary. For instance, the Locks tab does not appear when viewing properties of directories. If you are not logged in as a system administrator, and a database manager account was not created in Avstar NRCS, the dialog will appear like this:



The options in the dialog box appear gray, indicating they are for viewing only and cannot be altered. So, any user can view the traits of directories and queues in the Avstar database from a workstation.

## From the Avstar Console...

To get information about stories in your Avstar database, or to view a list of database traits for several directories or queues at once, you can use the `list` command at your Avstar console:



- `list d <folder/queue name>` provides information about directories or queues
- `list q <queue name>` provides information about stories in queues.

Both of these commands have a “verbose” option, such as `list d-v`, that gives you more detailed information. For instance, a verbose list, such as `list q-v`, includes read and write group information for each story in the queue. Read and write groups are explained in Chapter 6, “Groups.” Also, see Appendix G, “Managing Traits at Console,” for more information.

## Sending Output from the List Command to a Printer

Sometimes you may want a printout of database traits assigned to certain queues and directories. This can be done at the console.

To send the output from `list d` or `list q` to a printer, precede the command with `print` and the printer number. Use the same printer number you use when printing from an Avstar Workstation.

For instance, to send to printer 4 a list of the traits for the Rundowns directory and all the subdirectories and queues in that directory, type:

```
print 4 list d rundowns
```

The `print` command works with any variation of the `list` command or with any command that generates output on the console screen.

## Getting Information about Stories

The `list q` command lists story information for any of your system’s queues. The basic format of the command is as follows:

```
list q <queue name> [<record limit>]
```

In the `<record limit>` field, specify the number of stories from the queue you want to list.

For instance, to list the first three stories in `PEOPLE . CAROLYN . NOTES`, type:

```
list q people.carolyn.notes 3
```

A message similar to the following appears:

```
PEOPLE.CAROLYN.NOTES.SEARCH id=449889
rec quick index  LHDM-WObfpRmFf.id  time  modified-time
2  pm-chronology--DM-W-----457243  165   Jul 10 16:16:39 2000
3  pm-thumbnails--DM-W-----487595  163   Jul 10 16:21:17 2000
```

By default, the stories are listed in chronological order, with the oldest story first.

To list information for a particular story, use this format:

```
list qindex=<index value> q <queue name>
```

The *index value* is the value of the selected sort field of the story you want to list. This value is typically the text found in the title field, but you can set different fields as the index field. The quick index value must be a single word, with no spaces, and can be uppercase or lowercase.

For instance, to get story information for a story called `Nomad` in the queue `PEOPLE . ARLIN . NOTES`, type:

```
list qindex=nomad q people.smith.notes
```

A screen similar to the following appears:

```
PEOPLE.SMITH.NOTES id=449889
rec quick index  LHDM-WObfpRmFf.id  time  modified-time
3901bc-exp--nomad--DM-----420690  165   Jul 6 20:23:11 2000
```

The one-letter flags (LHDM-WObfpRmF) after the quick indexes provide current status information. The flags are:

L	Locked	b	Story's body (text) is edit locked
H	Held	f	Story's fields are edit locked
D	Duplicated	p	Story's production cues are edit locked
M	Modified	R	Read-only
-	-----	m	Mail
W	Wire	F	Floating
O	Ordered		

You cannot change any of these flags from the console, except the edit-locked status, which you can remove from a story with the `unbusy` console command. For instructions, see “Unbusy Stories and Queues” on page 5-53.

## Finding out Who Moved, Duplicated, or Killed a Story

To list the last person to move, duplicate, or kill a particular story in a queue, use this format of `list q`:

```
list qindex=<search word> q-mb <queue name> <record>
```

Since the `Index` field is typically the field containing the story's title (slug), it can be used as the `<search word>`.

The `<search word>` is a word from the `Index (sort)` field of the story. It must be a single word, with no spaces. It is not case-sensitive. The `<record>` is the numerical limit of stories provided in response to your `list` command—for instance, the most recent stories killed in the `Dead` queue.

To list this information for each story in a queue when you do not know a word in the `Index` field—the story's title—use this format:

```
list q-mb <queue name> <record>
```

The `b` in the command is optional. The difference between `list q-mb` and `list q-m` is:

`list q-m` lists begins at the oldest story in the queue  
`list q-mb` lists from the most recent material in the queue,  
such as the most recently killed stories in Dead.

You can also run the command without the `m` to see the date and time stories were moved, duplicated or killed. For instance, type:

**list q-b dead 5**

A screen similar to the following appears:

```
DEAD id=123231
rec  quick index  LHDM-WObfpRmF f.id time  modified-time
0      kyw-directors  ---M--O----- 3144900   Sep 6 09:51:58 2000
-1     a              ---M-----    16174611  Sep 6 09:15:06 2000
-2     sep 2000       ---M-----    3140930   Sep 5 17:34:19 2000
-3     008            --DM--O----- 313555600 Sep 5 16:49:24 2000
```

Here is an example of how to obtain the 5 most recently killed stories in dead:

**list q-mb dead 5**

A screen similar to the following appears:

```
DEAD id=123231
rec  quick index  LHDM-WObfpRmF f.idtime  user name
0      kyw-directors  ---M--O----- 3144900   palmer
-1     a              --DM--O----- 31358715  williams
-2     sep 2000       ---M-----    16174611  adbpurge
-3     008            ---M-----    3140930   ragusa
-4     008            --DM--O----- 313555600 ragusa
```

As shown in the example, some stories may be sent to the Dead queue by system processes, such as the automatic `dbpurge` (`adbpurge`). Lines without names are old versions of stories that were not written in the database by a user; for instance, they may have been put in the database through `txnet`.

Here is an example of how to get information for a story called “Camera” in the ASSIGNMENTS.MONDAY queue:

```
list qindex=camera q-m assignments.monday
```

A screen similar to the following appears:

ASSIGNMENTS.MONDAY id=14569				
rec	quick index	LHDM-WObfpRmF	f.idtime	user name
0	camera	--DM-----	16217274	williams

Williams was the last person to move, duplicate, or kill this story; `list q-m` does not make any distinction between these actions. Killed stories can reside only in the Dead queue, while duplicated stories will have the `D` flag on their listing, as shown in the previous example.



*If a utility programs, such as a server or link, moves, duplicates, or kills a story, its device number is listed in the `list q-m` or `list q-mb` display.*

When using the `list` command, long list results will scroll out of sight on the Avstar console screen. Since you may need to search through a long list of stories, such as 5000 in the Dead queue, you can redirect the output of the `list` command to a file in the database. For instance, the follow example redirects output to a user’s Notes queue.



*You must be in a superuser shell, to use this command.*

```
AVSTAR-A# sh
# list q-mb dead 5000 | doc pu people.p.palmer.notes
# <CTRL-D> (Hold Control key down and press D)
```

## Recovering a Killed Story

You can recover a story that has been killed—moved to and currently resides in the Dead queue—from any Avstar Workstation.

To retrieve a story to the database from the Dead queue, do the following:

1. Log in as a system administrator—that is, with a superuser account. This ensures you access to the Dead queue. On most systems, access to the Dead queue is restricted.
2. Navigate to the Dead queue in the Directory panel and open it by double-clicking on it.
3. Locate the story that you want to recover in the Dead queue, by scanning the list of stories displayed in the Queue panel for the story title (slug) or using the Find or Find All command.
4. Select the story or stories you want to retrieve by doing one of the following:
  - a. Click on the selector button located to the left of the story's row in the Queue panel. The entire row is highlighted when selected.  
-OR-
  - b. Move cursor to row and press Shift-Spacebar.  
-OR-
  - c. Click on each row while holding down the Control (CTRL) key to select multiple stories. The Shift key can be held down if you want to select all story rows between two mouse clicks.
5. Copy the selected story (or stories) to the new location by doing one of the following:
  - a. Click on and drag the highlighted selection to another queue in the Directory panel and release. The selected stories will be copied to the new queue location.  
-OR-

The Dead queue cannot be indexed, so do not use the Fast Text Search feature.

- b. Use the Copy and Paste buttons or Edit drop-down menu options to copy and paste the highlighted selection into a new queue location.
- OR-
- c. Use the Duplicate command to copy the highlighted selection to another location—particularly if the Dead queue is read-only.

## Changing Database Traits

You must be logged on as a system administrator—that is using an account with the superuser trait—or provide the database manager (`dbmanager`) password to change database traits of directories and queues. For more information on `dbmanager`, see “Creating a Database Manager Account” on page 4-36.

As the system administrator, you can alter database traits of a single directory (folder) or apply your changes to any subdirectories or queues in that parent directory as well.

When you change a directory’s traits from a workstation, the changes only affects the directory you selected, unless you specify otherwise. This is directly opposite to what happens when changing database traits at the Avstar console. See Appendix G, “Managing Traits at Console,” for more information.



*It is recommended you ensure that no users are working in a directory or queue prior to altering the database traits of that directory or queue.*

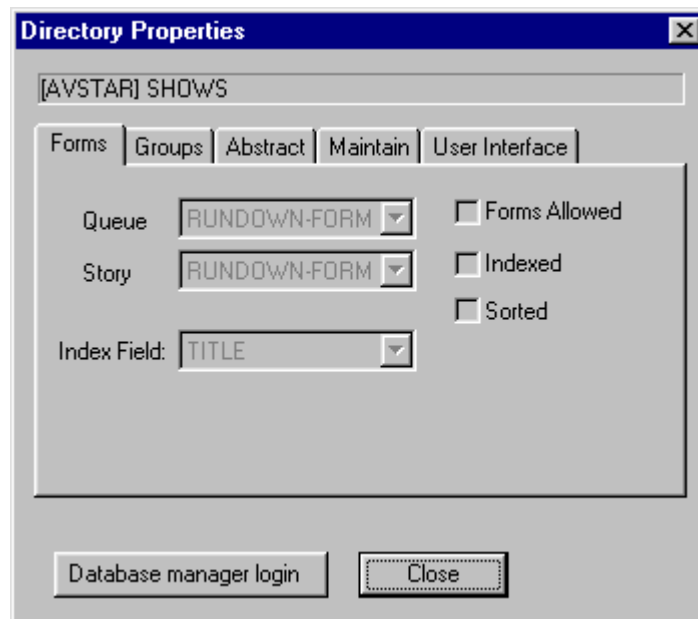
To change database traits, do the following:

1. Navigate to the directory you want to change in the Directory panel of the Avstar Workspace.
2. Right-click on that directory.

3. Select Properties from the pop-up menu. One of two things will happen, which will determine what you are to do next.
  - a. If you are logged on as a system administrator, the Directory Properties dialog box will appear. Go to step 5.

-OR-

  - b. If you are not logged on as a system administrator, the Directory Properties dialog box will appear in read-only mode—that is, all the fields in the dialog box will be gray, which indicates they are for viewing only. Go to step 4.
4. Click on the dbmanager login button (located at the bottom left corner of the dialog box) to gain access to change traits in the dialog box.

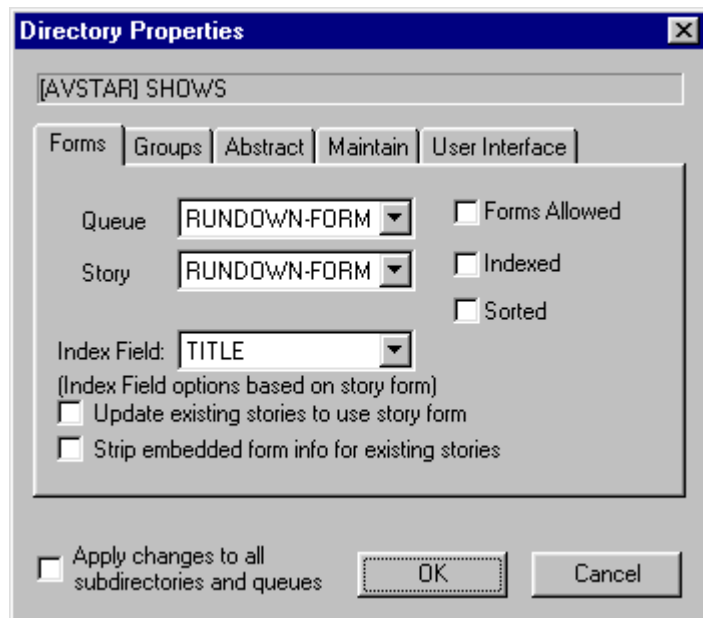




Avstar will prompt you for the database manager password. Fill it in, click OK.



The Database Manager Password dialog box disappears, and the dbmanager login button in the Directory Properties dialog box is replaced with a check box. Go on to step 5.



If you are altering the traits of a queue, you can skip step 5 completely because the Apply changes... check box does not apply and will not appear in the Queue Properties dialog box.



Selecting and/or unselecting checkboxes in the Directory/Queue Properties dialog box does not apply changes immediately. Only step 7 does that.

5. Do one of the following:
  - a. Click on the check box labeled Apply changes to all subdirectories and queues, if you want the changes you make to apply to all queues and subdirectories in the parent directory.  
  
-OR-
  - b. Do not select the Apply changes... check box, if you only want to change the traits of the chosen directory. The database traits are applied only to the directory selected when the Directory Properties dialog box opened.

*Selecting the Apply changes... checkbox does not apply the new settings at that point. It just indicates whether you intend to apply them to all subdirectories and queues. Also, when you apply your changes to all subdirectories and queues, those changes are not immediately apparent at all workstations. Users should log off and back on to see the changes.*

6. Make changes to the various traits as needed. These traits are explained in detail in the next section of this chapter.
7. Click OK to save changes & apply settings.

---

## Database Traits Summary

Assigning traits can be done from the Avstar Workstation as well as the Avstar console. For information on viewing and altering database traits from the console, see Appendix G, “Managing Traits at Console.”

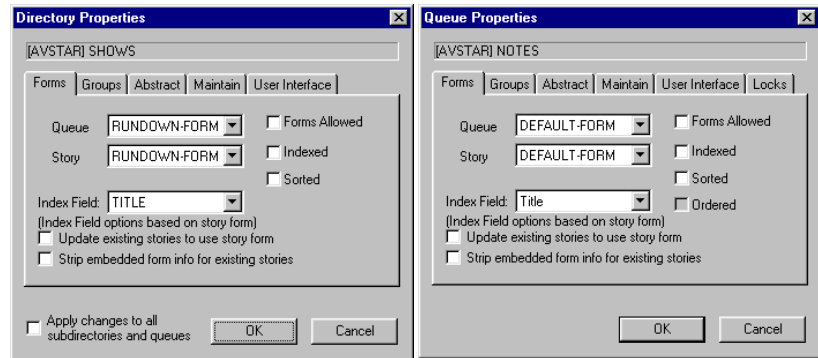
On the Avstar Workstation, the database traits are grouped together on various tabs in the Directory/Queue Properties dialog box. This section provides a detailed description of the dialog box, tabs and database traits. In some cases, traits offer a selection of options, such as what read group is assigned to a queue. These traits are usually shown as drop-down lists in the dialog box. In other cases, traits are either assigned to a queue or not—that is, the trait is “turned on” or “turned off.” These traits are usually shown as check boxes in the dialog box.

### Directory/Queue Properties Dialog Box

The appearance of the Directory/Queue Properties dialog box’s changes slightly, depending on whether you choose to view properties for a queue or a directory. First, the dialog box’s title bar will appear different, indicating that choice.

Other differences include checkboxes and tabs. For instance, the Apply changes... check box only appears in the bottom left corner of the Directory Properties dialog box. Also, the Ordered check box only appears in the right column of the Forms tab in the Queue Properties dialog box.

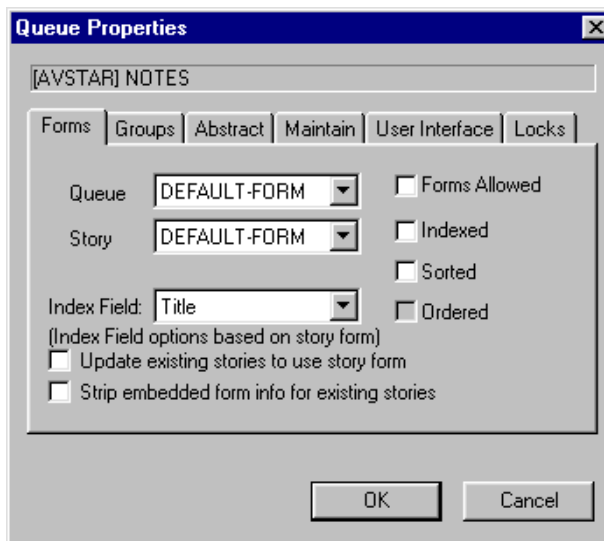
The number of tabs varies as well, depending on your choice of directory or queue.



The tabs are:

- Forms
- Groups
- Abstract
- Maintain
- User Interface
- Locks (This tab only appears for Queue properties.)

## Forms Tab



The Forms tab is unique because it is the only tab that allows access to certain items even when a user is connected to a local database, as opposed to the mirrored database on the system's servers.



*If you are connected to a local database on your workstation, you can still change the queue and story form selection using the drop-down lists. However, all other traits in the Directory/Queue Properties dialog box will appear gray, indicating access to them is read-only.*

- **Queue**
  - The Queue drop-down list, allows you to select a form used to display information in the Queue panel. The form defines what fields appear, which should be a sub-set of the fields used in the story form. A field included in the queue form that does not actually exist in the story form cannot be written to in the Queue panel. When !<none> is selected, no form is applied. This drop-down list is the equivalent

of the database trait (dbtrait), `qform`, at the Avstar console.

- **Story** – The Story drop-down list, allows you to select a form used to display information in the form fields of the Story Form panel. When `!<none>` is selected, no form is applied. This drop-down list is the equivalent of the dbtrait, `sform`, at the Avstar console.
- **Index Field** – The Index Field drop-down list, allows you to select a form field used if a queue is sorted (usually the Title field, also known as Slug field). This field also determines which form field the computer searches during a fast text search. Additionally, the cursor is placed on this form field by default when a user displays stories in a queue. This drop-down list is the equivalent of the dbtrait, `sortfield`, at the Avstar console.



*The optional fields in the Index Field drop-down list depend on the form selected in the Story drop-down list on the Forms tab. For instance, if you select a story form that only contains two fields, such as Title and Writer, then those two fields will be the only options listed in the Index Field drop-down list.*

- **Update existing...**– The Update existing stories to use story form check box, when selected, changes the story form assignment for previously existing stories within a queue. This check box is the equivalent of the dbtrait, `cform`, at the Avstar console.
- **Strip embedded...**– The Strip embedded form info for existing stories check box, when selected, removes embedded form traits from stories. For instance, queues with the Forms Allowed trait stamps the look of the story form into the story.

Assigning a different story form to one of these queues and selecting Update existing... check box will not affect the look of stories with the embedded forms. You would need to strip the embedded “look” from the story so it would then use the form assigned to the queue it is in. This check box is the equivalent of the dbtrait, `stripform`, at the Avstar console.



*The Update existing... and Strip embedded... check boxes are not database traits, but rather, they are used to apply current settings and/or changes in the dialog box at present. This means they will always appear unchecked when the dialog box opens, and they will not appear at all if the Directory/Queue Properties dialog box is opened in read-only mode.*

- **Forms Allowed** – The Forms Allowed check box must be assigned to all queues in the Forms directory. The forms will not work without this database trait applied. Additionally, this trait can be assigned to any queue in the database, but is usually only assigned to other queues that receive stories from other systems via `rx/tx` and then build forms for those stories, as needed. This check box is the equivalent of the dbtrait, `+f|-f`, at the Avstar console.
- **Indexed** – The Indexed check box, when selected, applies the Index trait. This trait is assigned to queues and directories that you want to be indexed by the Fast Text Search (FTS) server. This allows for quicker searching of the queue or directory. This check box is the equivalent of the dbtrait, `+index|-index`, at the Avstar console. See “Batch Indexing” on page 14-75 for more information.
- **Sorted** – The Sorted check box, when selected, applies the sort trait, which determines whether or not the stories in a queue will be sorted. Queues

with the sort trait are sorted by the form field you choose in the Index Field drop-down list. For instance, you may want to sort a rundown queue by the Page Number field, so when a user changes the numbering in the fields of that column, the rows automatically adjust to the numerical order. This check box does not start the sorting function. See “Starting the Queue Sort Function” on page 5-33 for more information. This check box is the equivalent of the dbtrait, `+s0 | -s0`, at the Avstar console.

The Ordered check box only appears on the Queue Properties dialog box, not the Directory Properties dialog box.

- **Ordered**
  - The Ordered check box is a unique check box, because it may appear as read-only, depending on the circumstance. It is provided to show you whether a queue is currently ordered. This is particularly helpful in identifying queues that have the sort trait, but are no longer being sorted because a user manually adjusted the order of the stories in the queue. If a sorted queue was manually ordered, the check box appears white and contains a checkmark, which you can remove if you want to reinstate the sorting feature. See “Starting the Queue Sort Function” on page 5-33 for more information. However, if a queue is not ordered at present, then the box will appear gray and empty. You cannot select this box to order a queue. See “Finding out Who Last Ordered a Queue” on page 5-48 for more information.

### Index Field/Story Form Compatibility Error Messages

A story form is a form that defines the fields displayed in the Story Form panel of the Avstar Workspace. The fields typically consist of important information about the stories stored in the queue, such as the title, writer’s name, and the dates the story was created or modified. Before Avstar’s Fast Text Search feature (FTS) can be used to



locate stories in a queue, the stories must be indexed by one of the fields located in the story form. Therefore, it is crucial that the field selected as the Index Field actually exist as part of the story form. For this reason, the system will check for compatibility between the settings of the Story drop-down list and the Index Field drop-down list. Depending on the list selections, various messages may appear.

When the index field is already defined based on the current story form setting, and a user selects a different form from the Story drop-down list, the system will check to see if the field chosen as the index field exists in the new story form. If it does not, a warning is issued. For instance, the Index Field drop-down list is set to a field, such as Audio-Time, and the Story drop-down list is set to the Rundown story form. At the current settings, there is no warning because the Audio-Time field exists in the Rundown story form. However, if the user changes the story form from Rundown to another form that does not have the Audio-Time field, the following message appears.

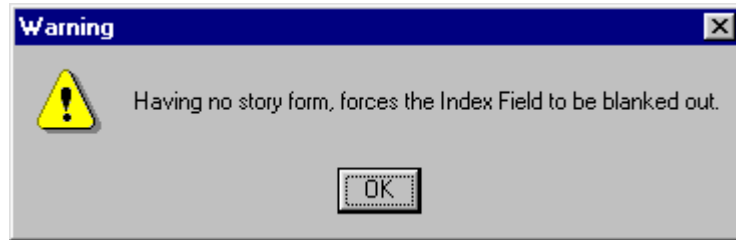


The Audio-Time field is no longer in the Index Field drop-down list. The user must select another field from the list as the warning message instructed. When the user clicks on the Index Field drop-down list, the system refreshes the list of options to display only those fields that exist in the currently selected story form.



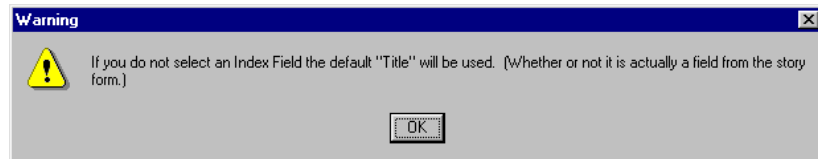
*Normally, pressing the ESC key after the Index Field drop-down list is open (and therefore, refreshed), will close the list, retaining the original field selection. However, if the original field choice does not exist in the newly chosen story form—such as the Audio-Time field in the previously mentioned example—the list closes without any selection made, in which case, the setting appears blank.*

If you select no story form—or set it to !<none>— then the system is forced to blank out the index field and issue the following warning message.

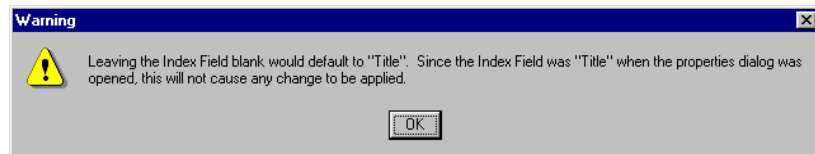


This means the Index Field drop-down list will be inaccessible and appear grayed out. When the index field setting is blank, the system will use the default field, which is the Title field (also called the Slug field).

When an index field is blank—or in other words, no field is selected—the following message appears.



If the index field drop-down list was already set to the Title field when the Directory/Queue Properties dialog box opened, and the user manually blanks out the setting, the following message appears.

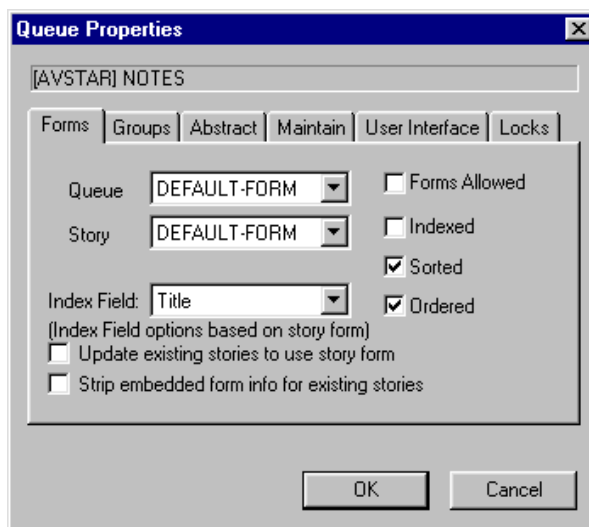


The above message indicates that by blanking out the index field setting, the default field, which is Title, is automatically applied. Since the default is the same field as the original field setting, no change occurs.

## Starting the Queue Sort Function

If someone orders a sorted queue, the sorting is disabled even though the trait is still applied to the queue. You restart the Queue Sort Function from the Avstar console or any Avstar Workstation. See “Starting the Queue Sort Function from the Console” on page G-26 for more information.

When a sorted queue is ordered, the Queue Properties dialog box will appear similar to this:

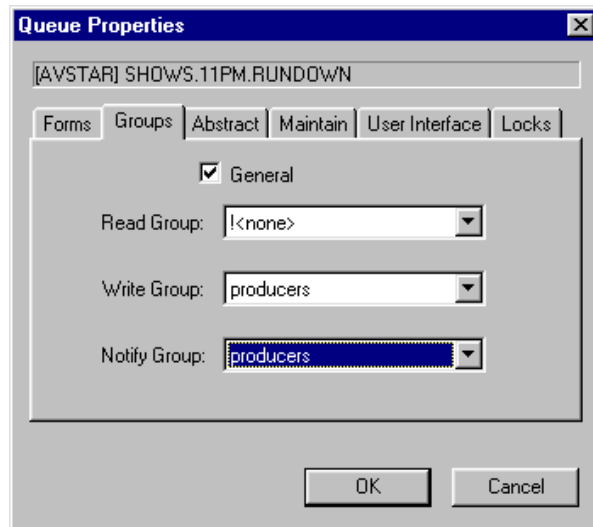


*You cannot use the Ordered check box to apply the Ordered attribute. In other words, when the box is not checked, it will appear gray, indicating that it is read-only. Ordering a queue is done by manually moving stories around within the Queue panel.*

To turn off the Ordered attribute and allow the queue to resume its original sorting function from the workstation, do the following:

1. Click on the Ordered check box, removing the checkmark.
2. Click OK to save the change.

## Groups Tab

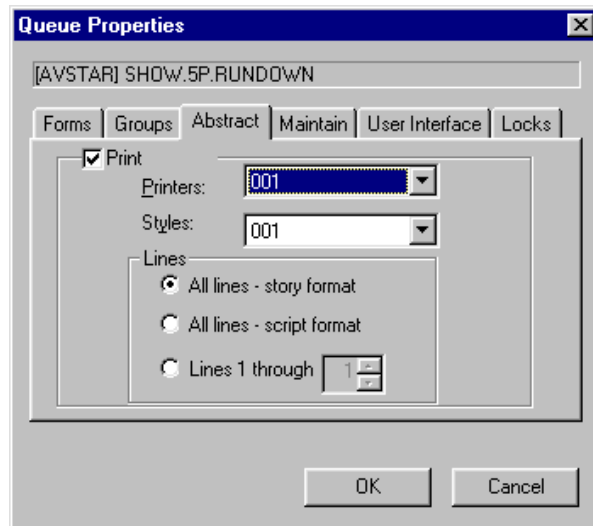


- **General**
  - The General check box, when selected, specifies that stories moved to the queue will retain their original security restrictions. For instance, this trait should be assigned to the Dead queue which normally allows unrestricted access to stories. This will prevent users from opening restricted stories once they are deleted from other queues—that is, moved to the Dead queue. See Chapter 6, “Groups,” on page 6-1 for more information. This check box is the equivalent of the dbtrait, +g | -g, at the Avstar console.
- **Read Group**
  - The Read Group drop-down list allows you to assign read access to a queue or directory to a group of users. Users who are not in the read group cannot see the directory or queue. When !<none> is selected, no group is applied; therefore, all users will have read access to the

queue or directory. The groups are not created here. See Chapter 6, “Groups,” on page 6-1 for more information. This drop-down list is the equivalent of the dbtrait, `rg`, at the Avstar console.

- **Write Group** – The Write Group drop-down list allows you to assign write access to a queue or directory. Users who are not in the write group cannot add or modify data in the directory or queue. Users cannot delete stories in any queue if they are not in the write group for the Dead queue. When `!<none>` is selected, no group is applied; therefore, all users will have write access to the queue or directory. The groups are not created here. See Chapter 6, “Groups,” on page 6-1 for more information. This check box is the equivalent of the dbtrait, `wg`, at the Avstar console.
- **Notify Group** – The Notify Group drop-down list allows you to specify what group of users is notified whenever stories are added to or modified in a queue. When `!<none>` is selected, no group is applied; therefore, no users will be notified of additions or modifications to the queue or directory. The groups are not created here. See Chapter 6, “Groups,” on page 6-1 for more information. This check box is the equivalent of the dbtrait, `ng`, at the Avstar console.

## Abstract Tab



When a story is added to or modified in an abstract print queue, the system automatically prints a portion of the story, as defined by the properties on the Abstract tab.

- **Print** – The Print checkbox, when selected, applies the abstract print trait to a queue or directory. The other aspects of this trait, such as printers and styles, cannot be applied until this trait is turned on. This check box is the equivalent of the dbtrait, a1, at the Avstar console.
- **Printers:** – The Printers: drop-down list allows you to select a printer the system will use to automatically print abstracts for the queue. This drop-down list is the equivalent of the dbtrait, ap, at the Avstar console.
- **Styles:** – The Styles: drop-down list allows you to select a style the system will use when automatically printing abstracts for the queue. This

drop-down list is the equivalent of the dbtrait, as, at the Avstar console.

- **Lines section**

- **All lines - story...**— The All lines - story format radio button, when selected indicates the queue has the abstract print trait, and the system will print all lines of new or modified stories in story format as an abstract from that queue.
- **All lines - script...**— The All lines - script format radio button, when selected, indicates the queue has the abstract print trait, and the system will print all lines of new or modified stories in script format—that is, including production cues—as an abstract from that queue.
- **Lines 1 through** – The Lines 1 through radio button, when selected, indicates the queue has the abstract print trait, and the system will print the designated number of lines of each new or modified story as an abstract from that queue.

### Uses for Abstract Printing

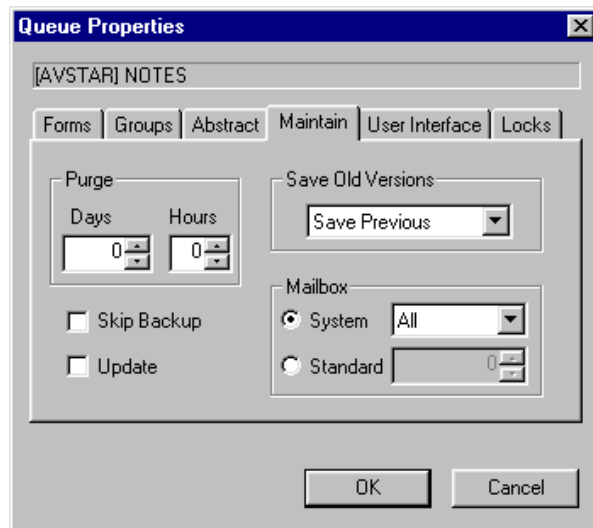
A common use for this feature is to print abstracts of wire stories on a specific topic. To do this, set up a wire keyword story to send to an abstract printing queue a copy of every wire story relating to the topic.

You can also use abstract printing to create a “paper trail” of changes made to stories in a particular queue. For instance, a producer may want a record of changes made to a show’s scripts. By making the queue where these scripts are written an abstract printing queue, the user can have the system print a record of every change made. Also, if the queue’s form includes a “modified by” field, the producer can get a record of who made each change.



If you edit a story in an abstract printing queue, your system prints an abstract of the new version of the story when you save the changes. To avoid printing an abstract, move stories to a queue that is not an abstract printing queue before editing them.

## Maintain Tab



- **Save Old Versions** – The Save Old Versions drop-down list determines how many old story versions are retained in each queue. The Save Old Versions drop-down list is the equivalent of the dbtrait, `save-|n|o|a`, at the Avstar console. Options include:
  - Save None – Retains no old versions of a story when a new revision is saved in the queue.
  - Save Previous – Retains the previous version of a story when a new revision is saved in the queue.



- Save Original – Retains the original version of a story when a new revision is saved in the queue.
- Save All – Retains all versions of a story when a new revision is saved in the queue.



*The Save Old Versions trait is queue-specific. For instance, a story is moved from a queue that saves all versions to a queue saving none. In this case, only the most recent version is moved to that queue. The old versions are sent to the Dead queue.*

- **Skip Backup** – The Skip Backup check box determines whether or not a directory or queue is left out of database backups. This check box is the equivalent of the dbtrait, `+x|-x`, at the Avstar console, and is also known as a skip flag.
- **Update** – The Update check box indicates whether or not the stories in a queue will be replaced as new versions are moved or copied to it. This check box is the equivalent of the dbtrait, `+u|-u`, at the Avstar console.



*The Update trait does not affect stories that are restored from tape backups. If you restore a story to a queue that already contains a version of that story, you will have two versions of the same story, even if the queue is assigned the update trait.*

See “Using Mailboxes” on page 14-14 for more information.

- **Mailbox section**

The Mailbox section does not apply to the e-mail feature of Avstar NRCS. These mailboxes are “signal carriers” by which utility programs, called servers, are notified to perform a pre-defined task. This section’s trait is the equivalent of the dbtrait, `mail`, at the Avstar console. There are two types of mailboxes:

- **System** – The System radio button and drop-down list are used to assign mailboxes reserved for

system functions, such as the keyboard and form checkers. Each queue or directory that needs a reserved system mailbox is assigned the correct one when the system is installed by iNews Customer Support personnel. Options include: All, Keyboard, Keyword, Distribution, and Group. See “Reserved Mailboxes” on page 14-16 for more information.

- **Standard** – The Standard radio button and drop-down list are used to assign mailboxes to queues. These are mailboxes used by server programs you can configure, such as action servers. The mailbox number assigned to the queue must match the device number of the server monitoring it. Valid mailbox numbers are one through 4096. See “Assigning a Mailbox to a Queue” on page 14-17 for more information.
- **Purge** – The Purge section allows you to set the recurrence schedule for purging a queue. The purge interval determines how old stories in a queue can get before they are purged. Every hour, your system removes any stories that are older than their queue’s purge interval and places the stories in the Dead queue. This process frees up space in your database for new stories. The Purge section’s Days and Hours spin boxes are the equivalent of the `dbtrait, purge`, at the Avstar console.

### Choosing Queues to be Purged

Any queue to which the system automatically sends stories should be purged regularly. For instance, queues in the Wires directory should be purged often, or else the database fills up with old wire stories. If

you set up a keyword story to send stories to a queue, ensure that queue is purged regularly. Otherwise, you risk running low on space.

Queues that do not automatically receive material usually do not need to be purged regularly. For instance, queues in the People directory are not normally purged automatically, since they usually only receive stories when users add them to their personal queues.

### Choosing a Purge Interval

Different parts of your database contain different kinds of stories, and the purge interval you set for each queue depends upon the kinds of stories in the queue. Some material, for instance, may need to be held only 24 hours while other material may need to be held three days—or not purged at all.

By choosing each directory or queue's purge interval based on the kinds of stories found there, you enable your system to remove old material that is no longer needed and make room for new material.

You can have purge intervals as long as 2729 days and 23 hours or as short as one hour. Additionally, you can set the purge interval to zero hours to turn purging off for a particular directory or queue. You can adjust the purge interval to suit the rate at which stories are added to the queue. In general, the faster stories are added to a queue, the shorter you should make the queue's purge interval. Old stories that you restore from tape are treated as if they were just created.



*If you notice an increase in the rate at which stories enter a particular directory or queue, you may need to reduce its purge interval. For instance, if you add another wire service to your system, the number of stories entering WIREs each hour increases, and you probably need to reduce the purge interval of WIREs to allow the system to keep up with this increase.*

### Matching Purge Intervals

When choosing a purge interval, pay attention to queues that are likely to contain copies of stories held in other queues. Copies of stories are really pointers that point back to the original story. The actual

story cannot be removed until all the copies have been purged. Otherwise, the copies would not have a story to which they could point.

Ensure queues that are likely to hold copies of a story have roughly the same purge intervals as the queue that holds the original.

In most systems, for instance, all wire stories are sent to `WIRES.ALL` and copies of each story are distributed to other queues in the Wires directory. This means that the system cannot purge a wire story until all the story's copies are also ready to be purged. To ensure that the story and its copies are ready to be purged at about the same time, the queues in the Wires directory are usually given similar purge intervals.



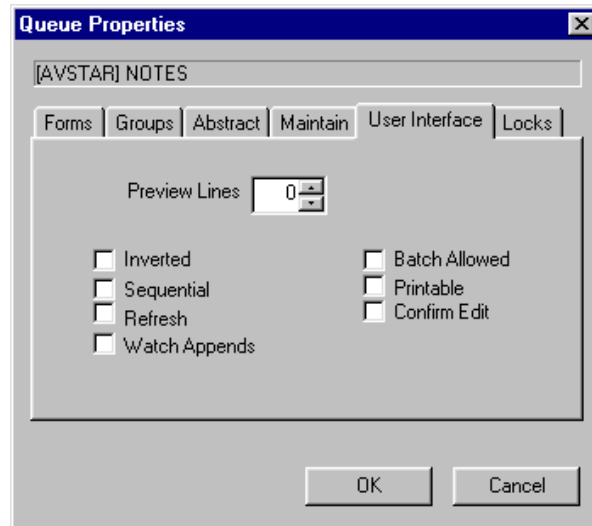
*If someone edits a copy of a story and then saves the changes, the system replaces the pointer with an actual story. Consequently, you do not need worry about copies that have been edited, since they no longer point back to the original story.*

### **Purge Intervals and the Purge Limit**

If the system detects that it is running low on space, it purges beyond each queue's purge interval, if necessary, to build up the free list. Your system profile contains a parameter called the *purge limit*, which prevents the system from purging more than a certain number of hours beyond each queue's purge interval.

In an emergency, your system purges as many hours beyond the purge interval as the purge limit allows. Queues that contain important information should have a purge interval at least five hours greater than the purge limit. This ensures that stories up to five hours old are never purged, even in a "low on space" emergency.

## User Interface Tab



- Preview Lines** – The Preview Lines spin box allows you to set a number of lines per story that will appear as a preview in the Queue panel. Usually, a queue will only show one line of information per story, similar to what appears in the fields of the Story Form panel. By applying the preview trait, users can also see a preview of each story’s text in the Queue panel, without having to open the entire story. A setting of zero will show the one line of information that is the standard; A setting of one will show that line plus one line of text, and so forth. This trait can be overridden by a user’s preferences. See “Preview Lines” on page 4-17 for more information. The maximum number of preview lines allowed is 22. This spin box is the equivalent of the dbtrait, `dis`, at the Avstar console.

- **Inverted** – The Inverted checkbox, when selected, will force the most recent stories in a queue to be displayed at the top. Otherwise, the most recent stories will appear at the bottom.
- **Sequential** – The Sequential checkbox, when selected, will force the directory or queue to list its contents in the order in which they were created. Otherwise, the contents are listed in alphabetical order. This check box is the equivalent of the dbtrait, +s | -s, at the Avstar console.
- **Refresh** – The Refresh checkbox, when selected, assigns the Refresh trait to a queue, so the system will begin automatically refreshing your screen when changes are made in the queue. This means when you are looking at a queue in the Queue panel, you will immediately see changes made to that queue by other users. This check box is the equivalent of the dbtrait, +refresh | -refresh, at the Avstar console.



*Use the Refresh trait only on important queues, like rundown queues that are often modified by multiple users simultaneously. To automatically refresh a queue, your system must spend a lot of time monitoring workstations where users are viewing that queue. Assigning the refresh trait to too many queues that are often accessed at the same time greatly increases the amount of work your system has to do and may severely degrade its overall performance.*

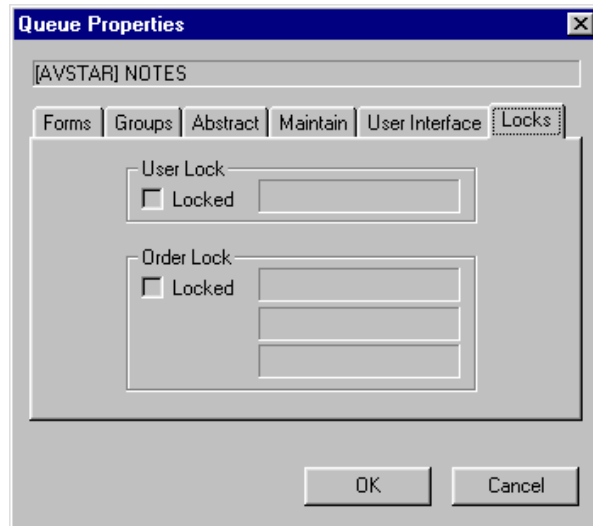
- **Watch Appends** – The Watch Appends checkbox, when selected, allows a queue to monitor incoming data for new stories sent by the wire service, appends them to the wire queue, and immediately displays them to users who have that wire queue open. While this trait can be applied to any queue in Avstar NRCS, it is crucial that it be assigned to queues that receive wire service data, such as the WIRES.ALL queue. This

---

check box is the equivalent of the dbtrait, `+w | -w`, at the Avstar console.

- **Printable**
  - The Printable checkbox indicates whether you can use the print command to print all stories in the queue with a single command. This trait is usually applied to rundown queues, such as `SHOWS . 5PM . RUNDOWN`. This trait does not limit or prevent the ability to print a single story (or script) in a queue. This check box is the equivalent of the dbtrait, `+p | -p`, at the Avstar console.
  
- **Confirm Edit**
  - The Confirm Edit checkbox, when selected, will instruct Avstar NRCS to prompt for confirmation before allowing a user to edit a story in the queue. This trait is typically assigned to queues in which users are likely to read but not change stories. It should not be assigned to queues with stories that are edited often. Doing so will needlessly slow down the performance of your system. This check box is the equivalent of the dbtrait, `+r | -r`, at the Avstar console.

## Locks Tab



The Locks tab is unique for two reasons: first, it only appears in the Properties dialog box for a queue not a directory, and secondly, it is a read-only tab. It is provided for informational purposes only. It cannot be used to alter the Lock/Unlock settings of a queue.

- **User Lock**
  - The Locked checkbox in the User Lock section indicates that a user has edit locked the queue. The User Lock field indicates the name of that user. Access to a locked queue is limited to a system administrator (with a superuser account) or someone who knows the correct key (password). The checkbox and field depict the current condition of the queue, so both are blank when the queue is not locked. However, you can find out the name of the last user to have locked the queue by going to the Avstar console.

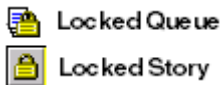


- **Order Lock** – The Locked checkbox in the Order Lock section indicates the queue is order locked at present, which limits who can rearrange the order of stories in the queue. The Order Lock fields indicates the user name and device which implemented the order lock, and when it happened. The checkbox and fields depict the current condition of the queue, so both are blank when the queue is not order locked. However, you can find out the name of the last user to have order locked the queue by going to the Avstar console.

## Locking and Unlocking

### Identifying Locked Queues and Stories

#### From the Avstar Workstation...



At the Avstar Workstation, you can identify a currently locked queue by the padlock that appears over queue icons in the Directory panel. As shown at left, a similar padlock appears on the selector buttons in the Queue panel when stories are locked. While the padlocks do not tell you who initiated the lock, you can find that out for currently locked queues at the Avstar Workstation. The information is displayed on the Locks tab of the Queues Properties dialog box. To access this dialog box, right-click on the queue in the Directory panel, and select Properties. See “Locks Tab” on page 5-46 for more information.

If a queue or story is not locked at present, but you want to know who last locked it, you must use the Avstar console.

## From the Avstar Console...

### Finding out Who Last Locked a Queue

To find out the last user to have locked a queue, go to the Avstar console, and use the following command:

```
list d-u [<queue name>]
```

For instance, to find out who last locked the `PEOPLE.SNOW.MAIL` queue, type:

```
list d-u people.snow.mail
```

See Table G-2 on page G-18 for more information on the list command one-letter flags.

The name of the person who last locked the queue appears in the `lockuser` column.

```
SRPl0-LI5UGQ5XWFi lockuser directory
Q-----I----N---- edmonds PEOPLE.SNOW.MAIL
```

In this case, `edmonds` appears to be the last person to have ever locked the `PEOPLE.SNOW.MAIL` queue. If the queue has never been locked, the name of the person who created the queue appears.



*The `list d-u` command lists the person who last locked the queue. However, if a name appears, it does not mean that the queue is currently locked.*

### Finding out Who Last Ordered a Queue

To find out who last order locked—that is, rearranged the order of—a queue and from which device the order was initiated, go to the Avstar console and use the following command:

```
list d-o [<queue name>]
```

For instance, to find out who last ordered the `CAST.MID.RUNDOWN` queue, type:

```
list d-o cast.mid.rundown
```

The name of the person who last ordered the queue appears in the `orderuser` column.

```
SRPlo-LIsUGQsXWFi orderuser device directory
QSRP-----U-Q---- williams 301 CAST.MID.RUNDOWN
```

In this case, Williams was the last person to order this queue. If the queue has never been ordered, the name of the user who created the queue appears. If no queue name is entered in the command, the order information for all folders and queues is displayed.



*The `list d-o` command lists the person who last ordered the queue. However, if a name appears in the `orderuser` column, it does not mean that the queue is currently order locked.*

To resume sorting in a sorted queue that has been ordered, see “Starting the Queue Sort Function” on page 5-33.

### Finding out Who Last Locked the Story

You can use the `wholockedit` command at the Avstar console to find out who last locked each story in a particular queue. Follow `wholockedit` with the name of the queue you want to investigate.

For instance, to find out who locked the stories in the `PEOPLE.BAKER.STORIES` queue, type:

```
wholockedit people.baker.stories
```

A screen similar to the following appears:

rec(3) modified by smith on Sun Jan 27 12:41:58 1993			
TITLE	PRESENTER	WRITER	MODIFIED
Air quality	brady	baker	Sun Jan 27 12:41

In the previous example, the story, Air quality, is the only one in the queue that is locked. Since Smith was the last person to modify it, you can assume that Smith either locked the story or knows its key.

## Types of Locks


There are four types of locks that can apply to either stories or queues: Edit lock, User lock, Order lock, Production lock. Not all the locks apply to both stories and queues. All four types are explained in the following sections.

### Edit Lock

An Edit lock is applied to a story to prevent multiple users from editing the story at the same time. The Edit lock can be applied manually by the user or automatically by the system, which is done when a user begins typing in the story.

#### Removing a Story's Edit Lock

The Edit lock is removed automatically when a story is saved, or when a user navigates to another location in the database. A user can remove the Edit lock (while remaining in the story) by saving the story or by doing one of the following:

- a. Type CTRL-E.  
-OR-
- b. Click the Story drop-down menu and select Edit unlock.  
  
-OR-
-  c. Click the Edit lock button on the Main toolbar (shown at left). The Edit lock is removed when the button no longer appears pushed in.

If a story is mistakenly left edit locked, it is considered to be in a busy state. You must *unbusy* the story before anyone can edit it. See “Unbusy Stories and Queues” on page 5-53 for more information.

## User Lock

A User lock is a lock that is manually (or purposefully) applied to a story or a queue by a user. Additionally, there are two types of User locks: Key lock and Easy lock.

- A Key lock is when a user locks the queue and applies a password, known as a key, to the queue. At that point, only system administrators and users who know the key can access the queue.
- An Easy lock is when a user locks the queue and applies their User ID as the key to the queue. At that point, only system administrators and users who log in using that User ID can access the queue.

When the user forgets the key or is unavailable to log in and unlock the queue, you must remove the lock before anyone can access the queue.

### Removing a Story's User Lock Without a Key

Before you unlock any story, find out who locked the story. You want to make certain that you do not unlock a story that is being used at the time. See “Finding out Who Last Locked the Story” on page 5-49 for more information.

To unlock a story without knowing its key, do the following:

1. Log in as a system administrator—that is, with a superuser account. You must be a system administrator to remove a lock for which you do not know the key.
2. Double-click on the story to open it in the Story panel. The system will prompt for the key.
3. Press Enter. The story will open without the key, because you logged in with a superuser account.
4. Be sure your cursor is in the Story panel.
5. Click the Tools drop-down menu.
6. Select the Unlock Story... option. A User Unlock Story dialog box will appear.

7. Click the Unlock button.
8. Save the story.

### **Removing a Queue's User Lock Without a Key**

Before you unlock (unbusy) any queue, find out who locked the queue. You want to make certain that you do not unlock a queue that is being used at the time. See “Finding out Who Last Locked a Queue” on page 5-48 for more information.

To unlock a queue without knowing its key, do the following:

1. Log in as a system administrator—that is, with a superuser account. You must be a system administrator to remove a lock for which you do not know the key.
2. Select the queue to unlock.
3. Click the Tools drop-down menu.
4. Select Unlock the Queue. The User Unlock Queue dialog box appears.
5. Click the Unlock button.
6. Click OK.

### **Order Lock**

An Order lock is automatically applied to a queue by the system. The system places an Order lock on each queue being ordered—that is, while someone is moving a story in the queue. Once the story is moved, the system automatically removes the Order lock. So, the lock only applies to the queue during the actual moving process. If you try to move a different story in a queue already being ordered, the system displays a busy message and temporarily denies order access to the queue.

If a queue is mistakenly left order locked, it is considered to be in a busy state. You must *unbusy* the queue from the Avstar console

before anyone can order it. See “Unbusy Stories and Queues” on page 5-53 for more information.

## Production Lock

A Production lock is similar to an Order lock in that it prevents multiple users from changing the order of a queue at the same time. The difference is that a Production Lock is manually applied to a queue by a user. Also, a queue will retain a Production lock until the user unlocks it, by doing one of the following:

- a. Click the Tools drop-down menu, then select Production Unlock.
- b. -OR-
- c. Navigate to another queue. Production Lock is disabled automatically.

If a queue is mistakenly left with a Production lock, it is considered to be in a busy state. You must `unbusy` the queue from the Avstar console before anyone can order it.

## Unbusy Stories and Queues

Whenever a story or queue cannot be unlocked from a workstation, you can remove the locks by going to the Avstar console and using the `unbusy` command.



**Always ensure a lock is invalid first. Determine whether the edit or order lock is not the result of someone actually editing a story or ordering a queue. Unbusy a story or queue currently in use can cause serious problems.**

To unbusy a story or queue, at the console, do the following:

1. Type the `unbusy` command
2. Follow it (on the same line) with the name of the locked queue or the queue that contains the locked story. For instance, if a locked

story was in `SHOW.SCRIPTS` or that queue was order-locked, unbusy it by typing:

**`unbusy show.scripts`**

3. Press Enter.

4. One of the following will happen:

- a. The `unbusy` command checks the queue for an Order/Production lock. If it finds one, the console displays the following prompt:

```
SHOW.SCRIPTS: orderlocked -- unorder? (n/y/q)
```

To unbusy the queue, type **y**; otherwise, type **n**. Do not remove an order lock that you have not already determined is invalid.

-OR-

- b. The `unbusy` command checks for an edit-locked story in the queue. If it finds one, it checks the workstation where the story was last edited. If anyone is logged in at that workstation, the console displays the story's form and prompts you to check with the person who is logged in at that workstation. For instance, if the story, Shootout, was last edited at workstation 11 and user, Smith, is currently logged in at workstation 11, the console displays the following:

```
WORKSTATION TAKE FROM MOVED STATUS  TIME
shootout 03-30 3044 APv1a1--hr Fri Apr 26 17:37 READY 0:47
did you check with smith who is at workstation 11 ?
workstation(11) unbusy ? (n/y/q)
```

To unbusy the story, type **y**, which will remove the edit lock. Otherwise, type **q** to quit.



**If a story is unbusied at the console while a user is editing it, when the user tries to save the story, the story is saved to the Dead queue, and the user gets an error message that states:**

```
Story save failed: Error: Story saved to dead.
```



# CHAPTER 6

## *Groups*

This chapter explains how to create groups in Avstar and use the system's group-related features to customize system usage. In some cases you will use the Avstar console, while in other cases, you can complete modifications from the Avstar Workstation. When possible, procedures are explained using the workstation, but some can also be accomplished at the console, with extensive use of the `gtraits` command. This command is similar to the `utraits` command and the `dbtraits` command. All three commands are explained in Appendix G, "Managing Traits at Console." This chapter also covers viewing and modifying information in the `SYSTEMS.GROUP` queue at the Avstar Workstation and `/site/system` file at the Avstar console.

Topics covered in this chapter include:

- Viewing Group Information
- Creating, Renaming, and Deleting a Group
- The Group Checker
- Creating or Modifying Multiple Groups in Interactive Mode
- Adding Members to an Existing Group
- Group Access and Usage Restrictions
- Group Traits for the Database
- Mail Aliases

## Overview

Avstar lets you categorize users by placing accounts belonging to people with similar needs into the same group. Organizing users in this way enables you to more easily customize your system to suit individual and small group needs. It also enables you to apply security restrictions at the group level.

For instance, you can:

- Restrict access to a particular queue so that only members of a certain group can use it
- Have the system notify a group of users when changes are made to a queue in which they are interested
- Send mail to a group name and have the system take care of the task of sending the mail to each individual in the group

You can create as many as 250 groups, and you can assign an unlimited number of users to each group.

## Viewing Group Information

### From the Console...

At the Avstar console, various versions of the `gtraits` command are used for viewing and modifying group information. You must be a console superuser before using any form of this command.

To get information for a list of all groups in the Avstar database, type:

```
AVSTAR-A# gtraits list
```

To get a list of all the members of a particular group, such as producers, type:

```
AVSTAR-A# gtraits list producers
```

To get a list of all groups a particular user belongs to, type the command followed by the user's ID, for instance:

```
AVSTAR-A# gtraits list danielmi
```

Sometimes, for security reasons, groups are assigned as group traits to directories and queues in the database. This information can be viewed from the Avstar console by using the `list d-v` console command, which lists each queue's assigned read, write, and notification groups.

To list group information for a queue, use this format:

```
list d-g <queue name>
```

To list all queues in the database that have a particular group assigned as their read, write, or notification group, use this format:

```
list rwng=<group name> d
```

## From a Workstation...

You can also view group membership and group assignment information from an Avstar Workstation. However, since groups are created at the console, the information you receive can potentially be out of sync with the information that is actually in the database. Therefore, use this method as a quick way to get information that you recognize may be down-level.

To view information about group memberships from a workstation, do the following:

1. Locate the `SYSTEMS.GROUPS` queue in the Directory panel.
2. Double-click the Groups queue to open it.

The Queue panel contains a list of the names of the existing groups with the first group name selected. The members of the group appear in the Story panel.

3. To view the contents of different groups, do one of the following:
  - a. Use the mouse to click a different group listed in the Queue panel.

-OR-

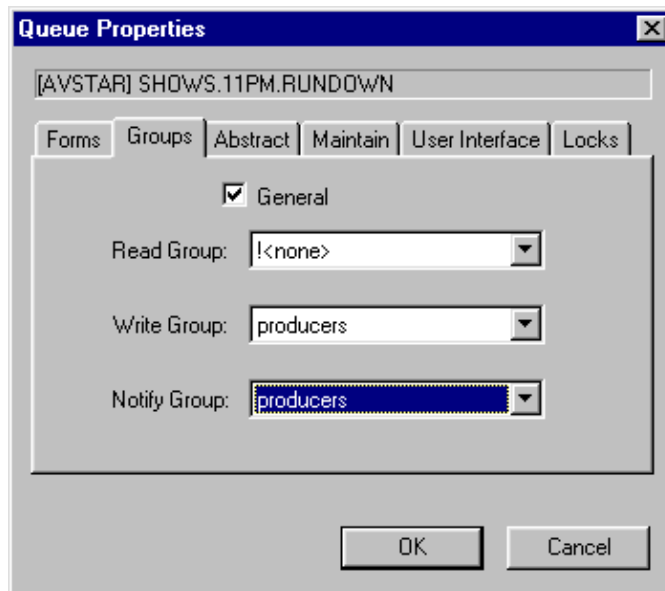
- b. Use the Up or Down arrow keys or the scroll bar on the right side of the Queue panel to move to another group.

-OR-

- c. Use Page Up or Page Down keys to scroll several groups (up or down) in the list at a time.

To view what group is assigned to a queue or directory for read and write access, or for notification purposes, do the following from a workstation:

1. Locate the queue or directory you want to know about in the Directory panel.
2. Right-click on the queue or directory.
3. Select Properties from the pop-up menu. The Queue/Directory Properties dialog box will appear.
4. Select the Groups tab. See “Groups Tab” on page 5-34 for more information.



---

## Creating a New Group

Creating a new group is a three-step process:

- Step 1 - Choosing a group name
- Step 2 - Enter the group name in the system
- Step 3 - Specify which users will be members of the group

The first part of the procedure must be done at the Avstar console. The second is done at the Avstar Workstation.

### Step 1 - Choosing a Group Name

Before you actually create a new group, you need to decide on the name of the group.

To choose a group name, follow these guidelines:

- Group names cannot be more than 20 characters long and cannot contain spaces. A group name longer than 20 characters will be truncated to 20 characters.
- You cannot use a name already used as someone's User ID.
- Some words are reserved by the system for special purposes, and cannot be used as names for groups. These words include: alias, all, group, and restricted.
- Choose a name that indicates the purpose or general makeup of the group, for instance, you may want to call the group that includes all your producers by the name "producers."

View the list of groups in the `SYSTEMS.GROUPS` queue to ensure the name you select is not being used for another group. If an existing group already uses a name, determine whether its members are the same as those users you want to assign to your group. The existing group may represent these users, so you can use it instead of creating a new one.

## Step 2- Enter Group Name in System

To enter the new group name in the Avstar database, do the following:

1. Verify that the new group name is not already used by typing the following kind of command at the console:

```
AVSTAR-A# gtraits list fieldreporters
fieldreporters is not a user or group name
```

In the example, the system response indicates that fieldreporters is not currently a group name. You should receive a similar response before proceeding.

2. Use `gtraits add` to enter the new group name in the system.

```
AVSTAR-A# gtraits add fieldreporters
```

## Step 3- Specifying Members of New Group

To specify members of the new group at an Avstar Workstation, do the following:

1. Open the System folder from the Directory panel.
2. Open the Groups queue.
3. Do one of the following:
  - a. Click the File drop-down menu and select New Story.

-OR-

  - b. Press the Insert key.

In the Queue panel, a blank row appears in the group list, and a blank story appears in the Story panel.



*When your Avstar Newsroom Computer System was installed, membership lists for several groups common to newsrooms were placed in the Groups queue. To make it easy to maintain, each list was placed in a separate story. Continue this convention to organize groups. You can put more than one group's membership list (or all of them) in a single story.*

4. Type the name of the group, such as `fieldreporters`, in the Title (Slug) field of the Queue panel or in the corresponding field of the Story Form panel.
5. Press Enter.
6. Click inside the Story Text panel and type the group name and membership list in this format:

```
group fieldreporters
user-ID user-ID . . .
user-ID user-ID . . .
. . .
```
7. Click the File drop-down menu.
8. Select Save Story.

This procedure creates a story, stored in `SYSTEM.GROUPS`, that bears the group name and contains the membership list for that group. The system will refer to the story anytime its group is applied to security measures or other system features.

## The Group Checker

When you save your changes to a group, the system automatically runs the server program known as the group checker, which looks for errors in the stories in `SYSTEM.GROUPS`.

The group checker may take a minute or two to process. When it finishes, the system sends you one or more messages describing the results. The system alerts you to the fact that you have received a message by an audio tone and a flashing Message bar button in the menu bar.



Click the Message bar button to read the messages from the group checker. The name, `grpcheck`, appears in the From field. The messages appear in the Message field.

If the group checker finds no errors in the story you have created to list the members of the new group, you get a `GROUPS story OK` message.



If the group checker finds an error, it sends a message indicating the story and the line in that story in which the error occurs. For a complete list of group checker messages, see “Group Checker Error Messages” on page 6-9.

In some cases, multiple errors are discovered, resulting in the issue of several error messages. To display the entire list of error messages sent to you, do one of the following:

- a. Click the History button.

-OR-

- b. Click the Communicate drop-down menu, select Messages, then choose Show History.

You can also copy or paste the contents of the group checker messages to another file.

If the errors are not serious, the last message is `Group story accepted`, with errors. The group checker applies changes that were not in error.

If there are serious errors, the last message is `Story NOT Okay`, indicating that the group checker could not use any changes because of errors.



The group checker examines whatever work you just completed along with everything else in `SYSTEM.GROUPS`. This means that if some of the error messages you see are not related to your changes, they are possibly the result of changes another user made in a different story in `SYSTEM.GROUPS`.

## Group Checker Error Messages

The following is an alphabetical listing of error messages you may see when you create or edit group membership lists or alias definitions. The group checker usually takes a minute or two to completely process the stories in `SYSTEM.GROUPS` and report any error messages.

Bad workstation device specification

You did not enter a workstation's device number or device name correctly. Usually, this happens because you did not use a closing brace (}) in the declaration.

Cannot open default aliases file

The group checker could not open an internal file that it uses to check alias entries. Call iNews Customer Support.

Cannot open new aliases file

The group checker could not create a new aliases file to reflect the changes you made. Call iNews Customer Support.

Cannot save old aliases file

An internal error occurred. Call iNews Customer Support.

Duplicate group or alias name

You tried to create two groups or aliases with the same name.

Failed to open queue

Due to an internal error, the group checker was unable to open `SYSTEM.GROUPS`. Call iNews Customer Support.

Failed to open story

Due to an internal error, the group checker was unable to open one of the stories in `SYSTEM.GROUPS`. Call iNews Customer Support.

Group or alias word missing. Skipping text

You did not begin a mail alias definition with the word, alias, or a group membership list with the word, group. This is followed by the line number where the group checker expected to find alias or group.

`GROUPS` story accepted, with errors

Errors appear in the group story, but none are serious. The group checker will use the entries that do not have errors. The entries that have errors are ignored.

`GROUPS` story NOT OK

Serious errors appear in the group story, and the group checker cannot use it.

`GROUPS` story OK

There are no problems with the group story.

Ignoring words following alias name

Any words you include on a line after the name of the alias you are defining are ignored.

Ignoring words following group name

Any words you include on a line after the name of the group you are defining are ignored.

Improper use of reserved word

You cannot use a reserved word, such as alias and group, as a group or alias name.

Internal groupchecker error

Some undefined error occurred while the group checker was running. Call iNews Customer Support.

Invalid name follows word "alias"

---

You added a nonexistent user to an alias or entered a user's name incorrectly. Check the alias and remove or correct the user name.

Invalid name follows word "group"

You added a nonexistent user to a group or entered a user's name incorrectly. Check the group's membership list and remove or correct the user name.

More than 50,000 alias names created

The system created many pseudo-alias names to break up individual aliases into lists of 1000 characters or less. Call iNews Customer Support.

Missing alias name

You did not follow the word, alias, with the name you want the alias to have.

Missing group name

You did not follow the word, group, with the name of a group.

Name already used as alias name

You created a group with the same name as an alias already defined in the story or queue.

Name already used as group name

You created an alias with the same name as a group already defined in the story or queue.

No groups or aliases found

All stories in SYSTEM.GROUPS are empty—they contain no aliases or groups.

Not a workstation device

You included something in braces ({ }) that is not a workstation device name or number. The message is followed by the name you tried to include as a workstation.

Not a user or workstation

You defined something that is not a recognized user or workstation as a member of a group.

Recursive group membership

You defined a membership list that created a recursion error. See “Group Access and Usage Restrictions” on page 6-20 for more information.

User name used as group or alias name

You cannot give a group or an alias the same name as an existing user.

## Renaming a Group

To rename a group in the Avstar system, you must complete two steps:

- Step 1 - Change the group name in the system’s database using the `gtraits` command at the Avstar console.
- Step 2 - Change the group name in the membership list story in `SYSTEM.GROUPS` at the Avstar Workstation.

### Step 1- Change Group Name in System

To change the name in the system’s database, do the following at the console:

1. Type the `gtraits r` command, which has this syntax:

```
gtraits r <old-group-name> <new-group-name>
```

For instance, to change the group name “producers” to “5pmproducers,” type:

```
AVSTAR-A# gtraits r producers 5pmproducers
```

```
Renamed producers to 5pmproducers.
```

### Step 2- Change Group Name in SYSTEM.GROUPS

To change the name in the membership list story, do the following:

1. Locate the group's membership list story in `SYSTEM.GROUPS`.
2. Modify the name of the group, which appears in the the title field of the queue and story form, and on the first line of the story.
3. Change the group name if it appears as a member in any other membership lists.



*If you do not do step 3, the next time you make a change in that story, the group checker warns you that the membership list uses an invalid group name. See “The Group Checker” on page 6-7 for more information.*

## Deleting a Group

To delete a group in the system, do the following:

1. Delete the name in the system's database at the console using the `gtraits d` command, which has this syntax:

```
gtraits d group-name
```

For instance, to delete the group “5pmproducers”, type:

```
AVSTAR-A# gtraits d 5pmproducers
```

```
Marked 5pmproducers for deletion.
```

This first step is a two-stage process. The `gtraits` command you type marks the file for deletion. The file is actually deleted the next time the system runs the `dbpurge` process, which it does at 15 minutes past every hour.



*You cannot use any `gtraits` commands on a group that is marked for deletion but still waiting to be deleted.*

2. Delete the group name and its membership list story in `SYSTEM.GROUPS`.
3. Delete the group name if it appears as a member in any other membership lists.



*If you do not do step 2, the next time you make a change in the queue, the group checker spots the deleted group's membership list and warns you that it uses an invalid group name. If you do not do step 3, the next time you make a change in that membership list story, the group checker warns you that the membership list uses an invalid group name. See "The Group Checker" on page 6-7 for more information.*

## Creating or Modifying Multiple Groups in Interactive Mode

The `gtraits` console command has an interactive mode in which you can execute group-related commands without entering `gtraits` each time. Use this mode when there are a number of `gtraits` commands you want to enter in succession.

To enter the `gtraits` interactive mode:

1. Become a superuser. See "Becoming a Console Superuser" on page 3-2 for more information.
2. Type:

```
AVSTAR-A# gtraits i  
>
```

The `gtraits i` command replaces the normal system prompt with an angle bracket (`>`) to indicate that you are in interactive mode. At this prompt, you can enter any `gtraits` command, such as `changegroup` or `add`, without typing the `gtraits` command for each operation.

For instance, to add the group "5pmproducers", type:

```
> add 5pmproducers
```

If you were not in interactive mode, the entire command line would be required, such as `gtraits add 5pmproducers`.

3. Type `quit` or `q` to leave interactive mode.

---

## Adding Members to an Existing Group

There are three possibilities for membership in a group. For instance, an individual user can be a member of a group, one group can be a member of another group (making all the users of the first group members of the second), and a workstation can be a member of a group.

### Users as Members of a Group

You must be at an Avstar Workstation to change group membership.

To add individual users to an existing group, do the following:

1. Open the System folder from the Directory panel.
2. Open the Groups queue.
3. Locate the story or stories in your `SYSTEM.GROUPS` queue containing the group membership list you want to add or modify.
4. Open that story in the Story panel.
5. Type in the user ID(s) you want to add to the group.
6. Click the File drop-down menu.
7. Select Save Story.
8. Verify the approval of the story from the messages sent to you from the system's group checker server program.

### Groups as Members of Other Groups

In addition to adding individual users to groups, you can add an entire group to another group. The members of the first group become members of the second group. The order in which you define groups in your `SYSTEM.GROUPS` queue is not important; however, care should be taken to avoid recursion.

## Avoiding Recursion

When you make groups members of other groups, do not create a membership list that contains a circular reference, also called *recursion*. The following is an explanation of recursion in a membership list.

When the group checker examines a group's membership list story, it builds an internal list of the group's members. If one member is in another group, the group checker must determine the members of the second group and add them to the first group's internal membership list.

In Figure 6-1, Group B is a member of Group A. When the group checker evaluates the membership list for Group A, it creates an internal membership list for Group A that contains users Fujitano, Clancy, Clancy, B, Meyer, Rosario, Chen, Reyes, and Smith. This example is not recursive and causes no problems for the group checker.

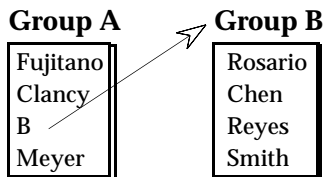


Figure 6-1 No Recursion

Recursion occurs when the group checker cannot resolve memberships because one group in the chain refers to another group higher up in the chain. Figure 6-2 demonstrates a case of recursion. Group B is a member of Group A, but Group A is also a member of Group B.

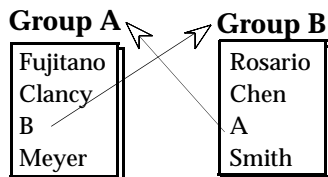


Figure 6-2 Recursion



The group checker cannot create internal membership lists for these groups. When it evaluates Group A, it sees that Group B belongs to Group A and tries to add B group's members to A group's internal list. However, one of the Group B members is Group A, which the group checker has still not resolved. The group checker cannot proceed.

If you see a recursion error message, examine your membership lists for incidents like this and remove the recursive reference. In the previous example, either remove Group B from the membership list for Group A or Group A from the membership list for Group B.

To check for recursion at the Avstar console, type:

```
AVSTAR-A# grpcheck -v system.groups
```

The group checker displays the title of the story in which it finds an error and a description of the error. For instance, after typing the `grpcheck -v` command, the following output shows a story with the title `group_249` that has recursive entries in its membership list:

```
grpcheck: 09:09:13 [CONSOLE] [group_249] Recursive group membership
a->b->a
grpcheck: 09:09:13 [CONSOLE] [group_249] GROUPS story NOT OK
```

## Workstations as Members of Groups

Suppose you have a workstation that is used by several staff members, all of whom are producers. When they use this workstation, they need access to the queues that members of the producers group normally have access to. When they are using other workstations, they do not need the special producer privileges.

You can grant the producer workstation the security permissions granted to the producer group by adding it as a member to the group called producers in `SYSTEM.GROUPS`.

When a user logs in at a workstation, the system ordinarily combines any system permissions the user already has with permissions the workstation may have. For instance, a user belonging to the writers

group who logs in at a workstation assigned to the producers group would have access to directories and queues accessible to both groups.

System permissions that apply to workstations are assigned using the `security` parameter that is set in the `/site/system` file. The security parameter in this file is either OR or AND. OR security uses the security level set for *either* the user *or* the workstation. AND security uses the security level set for *both* the user *and* the workstation. For more information on how to edit the `/site/system` file, see Chapter 10, “Ed, the Line Editor.”



**Using session numbers in groups will provide proper security only if you have dedicated resources locking down specific PCs to specific resources in the configuration file. See “Group Access and Usage Restrictions” on page 6-20 for more information.**

## Combining Permissions

Combining group and workstation permissions enables you to choose to apply additional security to your system, or less security:

- For additional security, you can specify that users at a particular workstation have membership in *both* human *and* workstation groups to perform certain actions.
- For less security, you can force the system to check only whether a user at a particular workstation is a member or *either* the human *or* the workstation group.

### Being More Restrictive

To be more restrictive about the permissions granted to users, include the `security` parameter in your system profile and assign it the value “and.”

For instance, you may use this security level for read permissions on a queue that should be read only by producers that are *both* members of the producers group *and* are sitting at the producers workstation.

### Being Less Restrictive

Omitting the `security` parameter from the system profile or assigning it a value of “or” indicates that a user on a workstation is considered to be in a certain group if *either* the user *or* the workstation is a member of the target group.

### Setting an Automatic Timeout

Either type of security described previously works well only if users log out from their workstations when they leave their desks. Otherwise, anyone with access to the workstation where a user logged in can take advantage of that user’s or workstation’s permissions.

To prevent this, you can set your workstations to automatically log out after a certain period of inactivity.

To do that, you need to edit one or more special timeout parameters in the system profile—that is, the `/site/system` file—at the Avstar console. See Chapter 10, “Ed, the Line Editor,” for more information on how to edit the `/site/system` file.

There are two types of timeout parameters:

- Idle timeout
- Login timeout

### Modifying Idle Timeout

The most useful kind of automatic timeout is the idle timeout, which logs out a workstation if no activity has taken place on it in a specified length of time.

To change this value for local workstations (those that are most likely within the newsroom), modify the `localtimeout` value in `/site/system`.

To set the idle timeout value for dial-up workstations, modify the `remotetimeout` value in `/site/system`.

### Modifying Login Timeout

You can set certain types of workstations to disconnect a user who does not log in within a certain length of time.

For network workstations and DOS PCs, modify the `netlogintimeout` value in `/site/system`.

To set this parameter for dialup workstations, modify the `logintimeout` value in `/site/system`.

### Timeout Value Settings and Format

All four timeout parameters accept values in the `mmm:ss` (minutes and seconds) format. The following table lists the parameters, the types of workstations they affect, and their maximum and default values.

Parameter	Workstation types	Maximum	Default
<code>localtimeout</code>	Serial, network, PCs	540:00	00:00*
<code>remotetimeout</code>	Dial-up	540:00	00:00*
<code>logintimeout</code>	Dial-up	1092:15	00:60
<code>netlogintimeout</code>	Network, PCs	1092:15	00:60

\* Disables timeouts

## Group Access and Usage Restrictions

Avstar is designed to be used by a large group of people, ranging from temporary writer interns to technical producers. To ensure the correct level of security on a system that is accommodating such a wide range of capabilities and responsibilities, restrict access to sensitive areas of the database to people with a need to access the information.

Avstar has security features that let you provide these kind of access restrictions. For instance, you can assign groups to a queue as a read and/or write group trait. By doing so, you can control which users can read and/or write stories in that queue.



**If you do not assign groups to a directory or queue as read and write group traits, the directory or queue is available to all users.**

## Access and Usage Examples

Here are some other examples how access is modified based on group trait assignments:

- If a user 's ID does not appear in the group(s) assigned as both read and write group traits for a queue, a user will be unable to create a story in that queue. In other words, the user will not have read and write permission to that queue.
- A user also needs both read and write permission to lock or unlock a queue.
- A user who has only write permission to a queue can copy or move a story into that queue; however, without read permission , the user will be unable to see the queue and its contents in the Directory panel.
- It is possible to change a directory's read or write group, and thereby modify the read-and-write permissions for all the stories in all the queues in that directory.
- Stories that you move or duplicate into a queue whose general trait is turned on, retain their original security. For instance, the Dead queue usually has this trait turned on so that stories moved there retain the read-and-write restrictions from their original queues.

## Group Traits for the Database

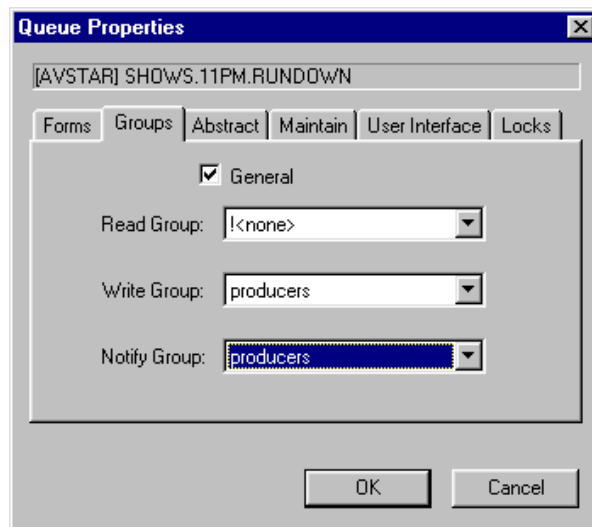
There are four group traits that can be assigned to queues and directories in the database:

- **General**
- **Read Group**
- **Write Group**
- **Notify Group**



*You must be a system administrator—that is, logged in with a superuser account—or know the database manager password to modify any trait in the Directory/Queue Properties dialog box. See “Directory/Queue Properties Dialog Box” on page 5-25 for more information.*

All four of the group traits can be assigned at the Avstar console and the Avstar Workstation. For procedures at the console, see “Groups” on page G-34 for more information. Procedures at the workstation, using the Groups tab in the Directory/Queue Properties dialog box, are covered in this chapter.



The General trait, when applied to a queue, means that stories moved to the queue will retain their original security restrictions, as set by the other three group traits. This will prevent any unintentional accessibility to stories that are moved from a highly secure queue to one that is widely accessible to users.

The other three group traits (Read, Write, and Notify) restrict who can read or write stories in a queue and indicate who is notified when stories are changed in it. Each of these group traits is explained in the following sections. Also, see “Groups Tab” on page 5-34 for more information.

## Read Group

A directory or queue’s read group specifies who can read stories in the queue. Users who are not in the read group for the directory or queue cannot see the directory or queue in the file structure displayed in the Directory panel.

To assign a group as a read group to a queue or directory, do the following:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the pop-up menu.
3. Select the Groups tab.
4. Select a group from the Read Group drop-down list. Only groups that are already created in the system database and defined in `SYSTEM.GROUPS` will appear in the list. When `!<none>` is selected, no group is applied; therefore, all users will have read access to the queue or directory.
5. Click OK to save settings.

To remove a group as a read group from a queue or directory, do the following:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the pop-up menu.
3. Select the Groups tab.
4. Select !<none> from the Read Group drop-down list. When !<none> is selected, no group is applied; therefore, all users will have read access to the queue or directory.
5. Click OK to save settings.

## Write Group

A queue's write group specifies who can add or modify stories in the queue.



*Users cannot kill stories if they are not in the write group for the Dead queue.*

To assign a group as a write group to a queue or directory, do the following:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the pop-up menu.
3. Select the Groups tab.
4. Select a group from the Write Group drop-down list. Only groups that are already created in the system database and defined in `SYSTEM.GROUPS` will appear in the list.
5. Click OK to save settings.



---

To remove a group as a write group from a queue or directory, do the following:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the pop-up menu.
3. Select the Groups tab.
4. Select ! <none> from the Write Group drop-down list. When ! <none> is selected, no group is applied; therefore, all users will have write access to the queue or directory.
5. Click OK to save settings.

## Notification Group

A queue's notification group specifies which users are notified whenever stories are added to or modified in the queue.

To assign a group as a notify group to a queue or directory, do the following:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the pop-up menu.
3. Select the Groups tab.
4. Select a group from the Notify Group drop-down list. Only groups that are already created in the system database and defined in `SYSTEM.GROUPS` will appear in the list.
5. Click OK to save settings.

To remove a group as a notification group from a queue or directory, do the following:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the pop-up menu.
3. Select the Groups tab.
4. Select !<none> from the Notify Group drop-down list. When !<none> is selected, no group is applied; therefore, no users will be notified whenever modifications are made to the queue or directory.
5. Click OK to save settings.

## Restricting Both Reading and Writing

You may need to restrict a queue so that one group of users can read and write in that queue, while another group can only read stories.

Suppose you want to restrict your Assignments directory. In most systems, a few people—mostly those at the assignments desk—need write permission to this directory. A larger number of users, such as writers and reporters, need to read, but not edit, stories in the Assignments directory.

The people who should have read-and-write permission for the Assignments directory come from different areas of the newsroom, so it is unlikely a group exists with just those users. However, you could set it up like this:

1. Create a group called assignments to represent users who need write permission for the Assignments directory.
2. Similarly, create a group called staff to represent users who need read permission.

3. Assign the staff group to the directory's read group trait and the assignments group to the directory's write group trait.

## Transferring Group Assignments

You may need to locate every instance where a particular group is assigned to a directory or queue and change that assignment so that another group is assigned to that directory or queue. Use this form of the `gtraits` console command:

```
gtraits transfer current-group-name new-group-name
```

Groups are marked for transfer, but no changes are made to any directories or queues until `dbpurge` runs. Both groups that you include in the `gtraits transfer` command must already exist.

## Hiding Queues and Directories

In addition to restricting access to various queues, you can use group access and usage restrictions to hide queues or directories by placing a strict read restriction on them.

A number of queues on your system probably have very tight write security to ensure that only certain users can create and edit stories in those queues. If other users do not need to read the stories in the queue, you may give the queue tight read security. This prevents the queue from appearing on unauthorized users' screens. Some examples of this are the Dead queue, a Suggestions queue, an Employee Evaluations queue, and so forth.



*All users that you want to have the capability to send stories to these queues need to have write access to the queue, but not necessarily read access. For instance, all users will need write access to the Dead queue so they can delete stories in other queues, which then moves them to the Dead queue. But you can limit who has access to the Dead queue by using the read group trait to restrict access to a certain group of users.*

Another example is the System directory, which is usually restricted so that only superusers can write stories there. You can hide this directory so that it does not appear in the main directory for normal users by setting its read group to a group that has no users. Because superusers can read everything in the database, they can still see the directory.

Many put the system administrator's name in the sysop group for e-mail purposes.

For this example, your system could use an empty group called sysop, which is then assigned as the read group trait for the System directory.

To set the System directory's read group to sysop:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the pop-up menu.
3. Select the Groups tab.
4. Select the sysop group from the Read Group drop-down list. If it does not appear in the list, it has not been created in the system database and defined in `SYSTEM.GROUPS`. See "Creating a New Group" on page 6-5 for more information.
5. Click OK to save settings.

## Mail Aliases

Some groups already in your system probably represent collections of people to whom you or other users want to send mail. For instance, you may want to send mail to all the producers in your newsroom. Producers probably already exist as a group in your system, and the mail system can use the producers group membership information to direct the mail to the right people.

You may also want to send mail to a group like the entire staff of your 5 p.m. newscast. That group may not already exist, and the need for

such a group may be related solely to mail delivery and not system security.

If you have such a need, do not create a group solely to meet the mail delivery objective. Use a mail alias, instead.

## Creating a Mail Alias

A mail alias is a name up to 20 characters long that represents a group of people who often receive similar mail. Each mail alias acts like a distribution list. This way, instead of sending mail to each user individually, you can send mail to the alias and the mail server distributes a copy of the mail story to each user on the group's membership list.

Like groups, mail aliases are defined in stories in the Groups queue in the System directory.

To create a mail alias, do the following at an Avstar Workstation:

1. Open the System folder in the Directory panel.
2. Open the Groups queue.
3. Click the File drop-down menu.
4. Select New Story.

A new blank Story panel appears. In the Queue panel, a blank entry in outline appears.

5. Type the name of the alias in the blank field of the Queue panel or in the corresponding title field of the Story Form panel.
6. Click inside the Story panel and type the alias name and membership list in this format:

```
alias alias-name  
user-ID user-ID group-name ...  
user-ID alias-name ...
```

An alias's membership list must begin with the word `alias` followed by the name of the alias and one or more lines that list user IDs, groups, or aliases that you want to include.

7. Click the File drop-down menu.
8. Select Save Story.

## Mail Aliases for Other Machines or the Internet

You can send mail to a user on another system connected to your system over the network. When you put a network mail address in a mail story's TO field, the mail server routes the mail to the correct address, which can be another machine on your Avstar system or anywhere on the Internet.

You can make it easier to use network mail addresses by assigning them mail aliases. Then, when you want to mail to someone who is not on a system connected to yours, use the mail alias.

To assign a mail alias to a network mail address, do the following at an Avstar Workstation:

1. Open the System folder in the Directory panel.
2. Open the Groups queue.
3. Click the File drop-down menu.
4. Select New Story.

A new blank Story panel appears. In the Queue panel, a blank entry in outline appears.

5. Type the name of the alias in the blank field of the Queue panel or in the corresponding title field of the Story Form panel.
6. Click inside the Story panel and type the alias name and membership list in this format:

```
alias alias-name  
network-address...
```

For instance, to assign the network address `jan@kbba_a.bba` to a mail alias called Jan, type:

```
alias Jan  
jan@kbba_a.bba
```

7. Click the File drop-down menu.
8. Select Save Story.





# CHAPTER 7

## *Keyboards and Macros*

Macros are time-saving routines you can assign to a programmable key(s) on your keyboard, which then can be evoked with one or two simple keystrokes. A single macro can be the shortcut to an entire command sequence.

A keyboard is a group of macros (programmable key definitions) that is stored in a description (story) file in the system directory. Each keyboard typically contains macros grouped according to specific job or task, such as a Producer keyboard.

This chapter explains how you can use macros and description files to customize keyboards within the following sections:

- Creating a New Keyboard Description Story
- Creating Macros
- Keyboard Checker Error Messages
- Assigning a Keyboard Macro Definition to a User
- Customizing Keyboards for VT/DOS Terminals
- For Users Who Switch Between Keyboards

## Understanding Macros and Keyboards

Macros let the user enter many characters and commands with a single key or key combination. You, as the system administrator, determine the function of each macro key. You can assign any command, sequence of commands, or plain text to a key or key combination. A set of macros is called a keyboard because it defines various actions that take place based on the keys pressed on an actual keyboard. A keyboard is usually created to contain a set of macros associated with a specific job, such as writer or producer.

For instance, suppose a writer in your newsroom frequently writes scripts for a particular reporter. You can assign the text in that reporter's usual sign-off to a macro key on the writer's keyboard. Then the writer could put the entire text of the reporter's sign-off in stories by pressing that key. This same key may be associated with a different macro, such as one that opens a rundown queue, if the user is a producer using a different keyboard, or set of macros.

Avstar keyboards are stored in queues in the `SYSTEM.KEYBOARDS` directory. See "Customizing Workstation Keyboards" on page 7-3 for more information.

You can assign macros to function keys, such as F1, F2, F3, and so forth, across the top of the keyboard and to the numeric keys on the numeric keypad, located at the right side of the keyboard. On Avstar Workstations (ASWS) you can use other keys known as state keys, such as Control (CTRL), in conjunction with the function and numeric keypad keys as shortcuts to entire command sequences. See "Creating a Macro" on page 7-5 for more information.

## Customizing Workstation Keyboards

An Avstar Workstation keyboard can contain more than 100 macros, representing possible states of the 12 function keys plus the 10 numeric keypad keys. For example, possible combinations for F7 are:

F7	CTRL-Shift-F7
Shift-F7	ALT-Shift-F7
CTRL-F7	ALT-CTRL-F7
ALT-F7	ALT-CTRL-Shift-F7



*Avstar NRCS will not recognize the Shift state key used in combination with the numeric keypad keys. For instance, {Shift- {CTRL- {kp9} }} is the same as {CTRL- {kp9} }.*

The keyboard, or set of macros, is actually a story that is saved in a specific location in Avstar NRCS. The macros are listed as text in the story. To create a keyboard containing a set of macros for Avstar Workstations, you create a story in one of the queues in the `SYSTEM.KEYBOARDS` directory.

The first story in the queue with a title containing the string `asws` is used as the keyboard macro definition story for Avstar Workstation. (The word `asws` is not case-sensitive.) The first story that does not have one or the other of those strings is used as the keyboard macro definition story for the legacy video terminal (VT) interface.

### Creating a New Keyboard Story

To create a new keyboard story:

1. Create a queue to hold the story in `SYSTEM.KEYBOARDS`. To do this from the Directory panel, Navigate to `SYSTEM.KEYBOARDS` and right-click on the folder, then choose New Queue.

The queue name must begin with a three-digit number. Append a hyphen immediately after the number, then a descriptive name for

the keyboard. If the number is less than 100, you must supply leading zeros.

For instance, to create a new keyboard named Producers, you could name the queue 005-PRODUCERS.



**The valid numerical range for queue names is 000-255 as shown in the follow examples:**

`system.keyboards.000-installation`

`system.keyboards.255-archivist`

**Queues with the number 256 or greater will not work.**

2. Open the newly created queue. To do this from the Directory panel, do one of the following:
  - a. Double-click on the queue icon
  - OR-
  - b. Select it and press Enter.
3. Create a new, blank keyboard story in the queue, by pressing the Insert key.

If an existing keyboard story is similar to the new keyboard description you want to create, you can copy it into the queue and modify it rather than creating a keyboard story from scratch.



*If writing macros for Avstar Workstation, the story's slug must contain asws in it.*

4. Create or modify the macros in the keyboard story. See “Creating a Macro” on page 7-5 for more information.
5. Save the keyboard story.



*The “key” mailbox is used by Avstar’s “keycheck” program, which checks for errors immediately after a keyboard story is saved. See “Keyboard Checker” on page 7-13 for more information. The “key” mailbox must be set on keyboard queues. The mailbox trait can be viewed and changed from the Directory Properties dialog box. See “Directory/Queue Properties Dialog Box” on page 5-25 and “Mailbox section” on page 5-39 for more information.*

## Creating a Macro

Keyboard macros begin with the “at” symbol (@) and are written in segments which make up what is called a macro definition. The segments include a Key Indicator, a Separator symbol (~), an Action, and an optional Comment. The segments must appear in the proper order for the macro to work correctly.

For instance: <Key Indicator> <Separator> <Action> <Comment>

The following is an example of a macro definition:

```
@{f4} ~ {alt gd} wires.all{enter};Go to wires.all
```

This macro allows the user to press F4 to navigate to the WIRES.ALL queue rather than completing the longer process—typing ALT-G-D to open the Destination dialog box, then typing **WIRES.ALL**, and pressing Enter—to do the same thing. The segments of the sample macro definition are explained in Table 7-1.

**Table 7-1 Macro Segments**

Macro segment	Function
@{f4}	The Key Indicator begins the macro definition line with the @ symbol. Then, {f4} indicates that the macro is invoked when a user presses and releases the key(s) defined within the braces—in this case, the F4 key. The braces {} are used to group letters together as one key or combination of keys.
~	The Separator is the tilde character (~) and it divides the key you are defining from the action that the macro is to perform.
{alt gd}wires.all{enter}	The Action includes a series of keystroke combinations. {alt gd} presses ALT, then types gd (for the Go, Destination drop-down menu options), and releases ALT. Then wires.all is typed in the text field in the Destination dialog box. And {enter} presses the Enter key and releases it.
;Go to wires.all	The optional Comment begins with the ; and provides a description of what the macro does.

## Adding Comments

Do not use the semicolon character in a macro. If you include a semicolon (;) in a description line in Avstar, everything following that character is treated as a comment by the system and is ignored.

A keyboard macro in Avstar NRCS can be as long as you want. All text that appears with an “at” symbol (@) at the start of a paragraph through to the end of that paragraph is considered to be part of a macro (except for comments as indicated by a semicolon).

If you create a macro longer than 80 characters, let the system wrap the cursor around to the next line. When you finish the macro, press Enter to start a new line for another macro. However, be sure that the next line starts with the @ symbol or a semicolon.



*If you end a paragraph and start another paragraph with anything other than an @ symbol or a semicolon, you will get an error message indicating an invalid key definition.*

## Assigning Macros to Keys

Keyboard macros for Avstar NRCS are written using the key names which are in the directory: `dictionary/site/dict/keymacros`. You can use alphabetic and numeric keys, and most punctuation marks in a macro. However, some punctuation marks are reserved for specific functions within a macro, as shown in Table 7-2. For instance, because the open brace ({), close brace (}), and tilde (~) characters have special meanings, you cannot use them as plain text in a macro. You can use the @ symbol in the action of the macro—to the right of the Separator, or tilde(~)—but not as plain text on the left-side of the Separator, where it indicates the start of a new macro definition.

Only certain keys can be defined as indicators for macros. On the Avstar Workstation, you can define macros for the function keys, and the numeric keypad keys, located at the right side of any standard PC keyboard.



*To use the numeric keys on the numeric keypad, Num Lock must be off.*

In most cases, you can define macros by combining the function or keypad keys with one or more state keys, such as ALT-F7. The other keys—listed in the second part of Table 7-2 as Edit, Arrow, and Miscellaneous keys—can appear in macros, but cannot have macros assigned to them.

**Table 7-2 Key Names**

Function keys	Keypad keys	State keys	
f1	kp0	shift	
f2	kp1	ctrl	
f3	kp2	alt	
f4	kp3		
f5	kp4		
f6	kp5		
f7	kp6		
f8	kp7		
f9	kp8		
f10	kp9		
f11			
f12			
Edit keys	Arrow keys	Miscellaneous	Reserved
insert	up	tab	@ (in Indicator)

Edit keys	Arrow keys	Miscellaneous	Reserved
home	down	esc	~
page up	left	backspace	{
page down	right	space	}
end		enter	;

## Predefined System Function Keys

Some function keys have predefined system functions, such as F1, which opens Avstar's Online Help System. These keys are provided as accelerator keys for common user functions.



**Redefining these predefined keys is allowed, but is not recommended. If you save a macro for a key that has a predefined function, the system displays a warning message stating that a reserved key has been redefined. See “Warning Messages” on page 7-14 for more information.**

The following table shows the standard predefined system function keys.

**Table 7-3 Predefined System Function Keys**

Function Key or Key combination	Predefined System Function
F1	Opens Avstar NRCS Online Help System
F2	Edit a field (or cell) in the Queue panel.
F3	Find Next
ALT-F4	Exits Avstar NRCS program
CTRL-F4	Closes a Workspace



<b>Function Key or Key combination</b>	<b>Predefined System Function (Continued)</b>
F5	Refreshes display in Queue or Story panel
CTRL-F5	Discard changes.
F6	Switches between Instruction panel (production cue) and Story Text panel
CTRL-F6	Toggles to next window.
Shift-F6	Moves between Story Text and Story Form panels.
F7	Opens Wire Urgent Workspace (Wire Priority queue)
Shift-F7	Opens Wire Alert History Window
F8	Toggles Message Toolbar on and off
ALT-F8	Communicate Message Reset
Shift-F8	Opens Message History dialog box
F9	Toggles Mail Workspace open and closed
CTRL-Shift-F9	Communicate Message Check

## The State Keys

The ALT, Shift, and CTRL keys are known as state keys, because their state affects what happens when another key is pressed, whether they are pressed or not. For instance, pressing F7, Shift-F7, ALT-F7, ALT-Shift-F7, and so on, can execute different macros.

To include a state key in a macro:

1. Begin with a { character to indicate that a key is being pressed.

2. Follow the { character with the name of the state key, such as ALT.
3. Since a state key does not do anything by itself, enter the name of the next key pressed along with it. Enclose this key in another set of braces { } if it is a function or numeric keypad key, such as F7 or kp1.
4. Close with a } character to indicate the release of the state key.



*Avstar will not recognize the Shift state key if used in combination with the numeric keypad keys. For instance, {Shift- {CTRL- {kp9} }} is the same as {CTRL- {kp9} }*

You can use state keys in the Indicator or Action sections of a macro: either as the indicator for defining the macro key combination or within the actions that the macro is to carry out. Here is an example:

```
@{alt{f8}} ~ {alt crl}
```

In the example, a state key is in the key indicator combination (ALT-F8) that executes the macro which contains another state key in the macro action itself (ALT-C-R-L).

Braces are used to group letters that are key names or key-combinations. For instance, {alt{f8}} is not the same as {alt f8}. In the first example, the ALT key is held down while the function key, F8, is pressed. In the second example, the ALT key is held down, while two separate keys, F and 8, are pressed in sequence.

The following table shows more examples of key combinations and how they appear in macros. The second combination shows how to write a macro using multiple state keys combined. It does not matter whether letters like these are in uppercase or lowercase in macros.

Key combination	Macro
CTRL-kp1	{ctrl{kp1}}
Alt-Shift-F7	{alt{shift{F7}}}
Alt-G-D	{alt gd}

## Using Plain Text in Macros

Besides using individual keys and key combinations in a macro definition, you can also have the macro enter plain text. This could be text you include in stories often or text that you enter in the fields of a dialog box in Avstar NRCS. For example:

```
@{ctrl{f9}}~{space} Roll tape - Sound up full
{space}{enter}
```

Whenever you include plain text in a macro, all spaces in the text are preserved. The case (lowercase or uppercase) is preserved as well.

## Repeating Macros

If there are actions performed at the workstation that require executing the same command or series of commands over and over, you can create a repeating macro that performs this action. Once invoked, a repeating macro executes at regular intervals until the user presses CTRL-Break or Escape.

To have a macro in an Avstar keyboard perform a repeating function, place the command `{repeat}` just before the actions you want the macro to repeat.

As an example, create a macro that makes it easier to browse wires and assign it to the F4 key. This macro takes the user to WIRES.ALL and then scrolls down one story at a time. It pauses briefly on each story, so the user can read the title and decide whether to read the story. The user stops scrolling by pressing CTRL-Break or Escape.

To create this macro:

1. Begin the macro with `@{f4}~` to indicate that it is for the F4 key.
2. Follow the Separator (~) with `{alt gd} wires.all {enter}` to open a window displaying WIRES.ALL.
3. Add the repeating portion of the macro, which moves the user down one line at a time at regular intervals. Begin by typing `{repeat}` to indicate that what follows is repeated. Then type `{down}` to move the cursor down one line in the queue.

The range for the `{pause}` option is 1 to 60 seconds.

4. Use the `{pause}` command to make the macro pause a few seconds on each story. Follow this command with the number of seconds you want the repeating macro to pause. To make the macro pause two seconds before repeating, type `{pause 2}`.

The macro should be one continuous line of text. Otherwise, allow the computer to wrap the text if it extends beyond screen margins.

The completed macro looks like this:

```
@{f4}~{alt gd}wires.all{enter}{repeat}{down}
{pause 2}
```

## Notes of Caution for Creating Macros

When creating macros—whether or not they are repeating macros—care should be taken. First, certain commands should never be used in macros.



**Do not use the Duplicate or Kill commands in repeating macros. Doing so raises the risk of accidentally killing the wrong stories or filling up the database and causing a “low-on-space” condition.**

Secondly, all macros should be created using steps only to a point where varying options are not a possibility. For instance, a system administrator wants a macro designed to open a story and type a specific production cue on a certain line, then save the story and open the next one in the lineup. The system administrator must keep in mind that the process for editing stories may vary for each user depending on user preferences or queue location. In other words, in some cases, a user may start to save a story and be prompted by a dialog box that requests confirmation. This confirmation box is a user preference that varies with each user. A similar confirmation dialog box may appear if a user is opening a story in a read-only queue. So, any macro must incorporate these possibilities or stop prior to them. Otherwise, the macro may hang up on an unexpected dialog box.



*To immediately stop a macro that is in progress for any reason—including one hung up on an unexpected dialog box—press the Escape (ESC) key.*

## Keyboard Checker

Whenever you modify and save macros in a keyboard description story, the system checks the keyboard and all its macros for problems that may prevent the macro from working properly. If it finds a problem, it sends you a message describing the error.

Avstar will also issue a warning if any predefined system function keys—that is, those keys reserved for Avstar system functions—are replaced with a macro. This warning can be ignored if you want to override the pre-defined system function keys with a macro. See “Predefined System Function Keys” on page 7-8 for more information.

As long as you get a `Keyboard ok` message, the description story can be used, but go back and fix any noted problems so the keyboard does what it is supposed to.

### Error Messages

Table 7-4 lists messages from the message dictionary that can appear after you save a keyboard description story.

**Table 7-4 Keyboard Checker Error Messages**

Error message	Explanation
Duplicate key description (M_KEYDUP)	You defined the same function key twice in the story; remove one of the definitions.
First key description does not begin with @ (M_KEYSTART)	You must use an @ symbol as the first noncomment character in the description story.
Invalid key number (M_KEYRANGE)	You tried to define a function key with a number that is not supported.
Keyboard description contains too many characters (M_KEYLONG)	This description story is too long. Shorten the macros by using command abbreviations, or deleting macros you do not use.

**Table 7-4 Keyboard Checker Error Messages (Continued)**

<b>Error message</b>	<b>Explanation</b>
Keyboard NOT usable (M_KEYBAD)	You must fix the errors in this description story before you can use it.
Keyboard ok (M_KEYOK)	You may use this keyboard, even if it has errors.
Missing key number separator (~) (M_KEYSEP)	You must follow the key number of an extended programmable key with a tilde (~).
Not enough key descriptions (M_KEYMIN)	You must include lines for all the standard VT keys. Include only the key's number on a blank line, if you don't want to assign a function to a standard key.
Warning: a key definition contains a repeating function (M_KEYREP)	One of your key definitions has a repeating function. If this is not what you want, edit the description.
Warning: badly placed @ exists in key definition line (M_KEYFUNKY)	A line in the story contains an @ symbol that is not the first character in the line or between two commands. If this is not what you want, edit the description.

## Warning Messages

Table 7-5 lists the messages that appear if the keys reserved for Avstar system functions are redefined.

**Table 7-5 Keyboard Checker Warning Messages**

<b>Reserved key</b>	<b>Warning message</b>
F1	Warning: "Help" key redefined (M_STDHELP)
F3	Warning: "Find Next" key redefined (M_STDFINDNEXT)
Alt-F4	Warning: "Exit" key redefined (M_STDEXIT)

**Table 7-5 Keyboard Checker Warning Messages (Continued)**

<b>Reserved key</b>	<b>Warning message (Continued)</b>
CTRL-F4	Warning: "Window Close" key redefined (M_STDCLOSE)
CTRL-F5	Warning: "Discard Changes" key redefined (M_STDDISCARD)
F5	Warning: "Refresh" key redefined (M_STDREFRESH)
F6	Warning: "Script Swap" key redefined (M_STDSSCRIPT)
CTRL-F6	Warning: "Next Window" key redefined (M_STDWINNEXT)
F7	Warning: "GoTo Priority Queue" key redefined (M_STDPRIORITYQUEUE)
Shift-F7	Warning: "GoTo Alerts History" key redefined (M_STDALERTSHISTORY)
F8	Warning: "Communicate Message Bar" key redefined (M_STDMESSAGEBAR)
Alt-F8	Warning: "Communicate Message Reset" key redefined (M_STDMESSAGERESET)
Shift-F8	Warning: "Communicate Message Show History" key redefined (M_STDMESSAGEHISTORY)
F9	Warning: "Communicate Open/Close Mail key redefined (M_STDMAIL)
CTRL-Shift-F9	Warning: "Communicate Message Check" key redefined (M_STDMESSAGECHECK)

## Assigning a Default Keyboard to a User

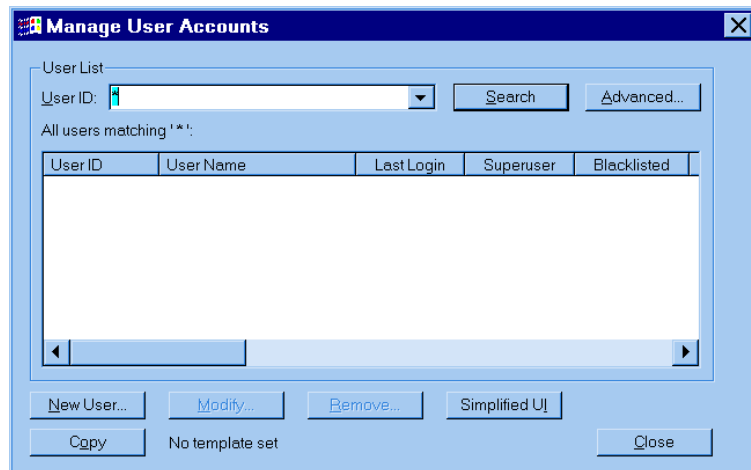
Users can select a keyboard—that is, a set of macros—to use at any time by using the Preferences option in the Tools drop-down menu. The system administrator can assign the default keyboard for a user, which appears when the user first logs on.

When you add a new user to the system, you may want to assign a keyboard as a default for the user at that time, according to the role the user plays in your newsroom. For instance, a new writer may get the “user” keyboard, a new producer would be assigned the “producer” keyboard, and so forth.

You assign a set of macros as a user’s keyboard by assigning the keyboard story containing those macros to that user. When the user presses a programmable key, the system looks at the user’s assigned keyboard story and executes the macro assigned to that key.

To assign a keyboard to a user as a default from an Avstar Workstation, do the following:

1. Click the Tools drop-down menu.
2. Select Options.
3. Select Users. The Manage User Accounts dialog box appears.





4. Do one of the following:

- a. Create a new user account by clicking New User.

-OR-

- b. Modify an existing user account by selecting a user and clicking Modify.

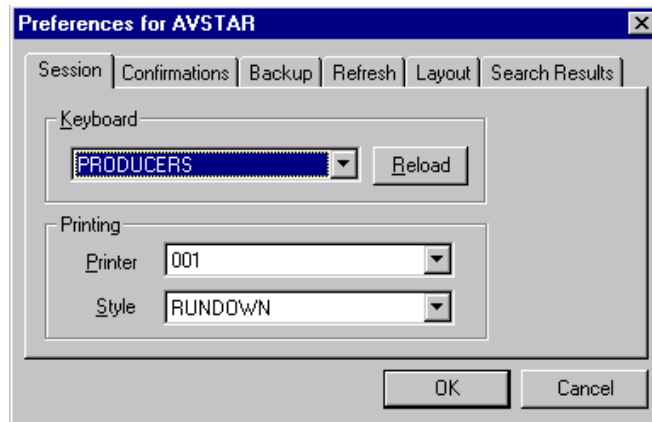
5. In the Modify User Account dialog box, click User Preferences.

If you clicked New User rather than Modify in step 4, the dialog box appearing in step 5 will be titled Add New User. See “Adding a New User Account” on page 4-26 for more information.



*Access to the Modify User Account or Add New User dialog boxes are restricted to certain users, such as system administrators and user managers—that is, users who know the umanager password. See “Modifying User Traits” on page 4-3 for more information.*

6. In the Preferences dialog box, click the Session tab, if not already selected.



7. Use the Keyboard drop-down list to select a keyboard for the user.
8. Click Reload.
9. Click OK to save the new keyboard assignment and close the Preferences dialog box.
10. Click OK to save the modified user account and close the Modify User Account dialog box.
11. Do one of the following:
  - a. Select another user to assign a keyboard to, and click Modify.
  - OR-
  - b. Click Close to close the Manage User Accounts dialog box.

The next time the user logs in, the computer will automatically assign the new keyboard. If the user is already logged in, he or she needs to reload the keyboard or log out and then log back in to use the new keyboard assignment.



*Users can choose a different keyboard than what is assigned to their user accounts at any time by doing the following: Click the Tools drop-down menu, select Options, select Preferences, choose the Session tab, pick a keyboard from the drop-down list, and click the Reload button.*

## Customizing Keyboards for VT/DOS Terminals

On a video terminal (VT), you can assign two macros to each programmable key. The first macro executes when the user presses the key; the second macro, an alternate macro, executes when the user presses the Command key (Enter on the numeric keypad) followed by the key. Almost all special keys on a VT are actually macro keys, including Save, New, Insert, Up Arrow, and so forth.

The macros for VTs are stored as sets, called keyboards. A keyboard is actually a story that is saved in one of the queues in the `SYSTEM.KEYBOARDS` directory. A user can have keyboard stories for both a VT/DOS terminal and ASWS in the same queue. The VT/DOS version of the keyboard must be the first “non-asws” story in the queue—that is, the first story that does not contain the string `asws`.



*VT keyboard stories must not include `asws` or `anws` in the title field (slug).*

The location of the macro keys varies slightly depending on which type of terminal keyboard you are using. Here is a sample keyboard story to show how to use macros to customize your keyboard for VT/DOS Terminals. The story is `SYSTEM.KEYBOARDS.005-SMITH`:

```
@{cursor up}~{cursor top}
@{cursor down}~{cursor bottom}
@{cursor left}~{go .}
@{cursor right}~{get}
@{script}~{script undo}
@{mail}~{mail reply}
@{seek}
@{find}~{replace}
@{video step}~{video define}
@{dup}~{move}
@{notes save}~{notes recall}
@{delete define}~{delete recover}
```

This story is a copy of the installation keyboard. The purpose of most macros is clear from the macro line. For example, in the first line of this story, you can tell that the first macro assigns the Cursor Up command to a key.

## Miscellaneous VT Macro Tips

A macro can be several lines long, so begin each macro with an “at” symbol (@) in the first column to indicate the beginning of a new macro. Everything between this and the next @ that appears at the beginning of a line is included in the macro.

Keyboard stories are limited to 2000 characters each. If you are building long macros, use the short form for each command, such as `cu l` rather than `cursor left`, to avoid exceeding the size limit.

### The Pause Command

If you want the system to wait a few seconds before it performs the next command in a macro, use the `pause` command. Follow this command with the number of seconds you want the system to wait. To have the system wait 10 seconds, for example, type `{pause 10}`.

### The Blank Command

If you have a lengthy macro and you do not want to see all the screen changes when it executes, use the `blank` command to make the screen go blank for all or part of the command sequence. The first `blank` command in a macro clears the screen. The message `please stand by` flashes at the top right of the screen. The commands that follow the `blank` command are executed, but the user sees only the `please stand by` message. The screen is redisplayed when the macro is finished or when another `blank` command is executed.



*By reducing the output, the macro runs much faster.*

## Repeating VT Macros

If you perform a procedure at the workstation that requires executing the same command or series of commands over and over, create a repeating macro. Once invoked, a repeating macro executes at regular intervals until the user presses a key.

To get a macro to perform a repeating function, place an @ character before the commands that you want the macro to repeat. Do not confuse this with the @ character you must put in the first column to indicate the beginning of the key definition.

As an example, add a macro to Smith's keyboard that makes it easier to browse wire stories. This macro takes Smith to WIRES.ALL and then scrolls down one story at a time. It pauses briefly on each story, so the user can read the slug and decide whether to read the story. The user can stop scrolling by pressing any key.

Because the sample keyboard description is extensive, replace one of the existing macros with this repeating macro. For example, if a user does not usually use a video step macro, replace it with a wire-browsing macro:

```
@{video step}~{video define}
```

This macro does the following:

1. Opens the keyboard story and moves the cursor to the last line. We want the first part of the macro to take the user to WIRES.ALL, so type @{go wires.all} on that line. Use @ to indicate the beginning of the macro.
2. Adds the repeating portion of the macro, which moves the user down one line at a time at regular intervals. Type @ to indicate that what follows is repeated. Type {cu d} for the cursor down command.

The system expects the @ character to follow a right brace ( } ) character. Otherwise, it sends a warning message informing you that the story has a badly placed @ character. This does not keep the system from using the story, but you may want to ensure that all your macros do what you want them to do.

3. Uses the pause command to make the macro pause a few seconds between cursor down commands, so that the user can examine each story. Follow this command with the number of seconds that you want the repeating macro to pause. To make the macro pause two seconds, type {pause 2}:

```
@{go wires.all}@{cu d}{pause 2}
```



*Do not use the Duplicate or Kill commands in repeating macros. This raises the risk of accidentally killing the wrong stories or filling up the database and causing a low-on-space condition.*

## Assigning VT Macros to Standard Macro Keys

Your system uses either VT220-style workstations or PCs running VT emulation software (a program that makes the PC act similar to a VT) — see Figure 7-1 and Figure 7-2.

There are 24 keys on a VT100, which Avstar uses as macro keys. The PC extended keyboard has only 22 of these. Most systems now use VT220-style keyboards or PCs with extended keyboards, although some VT100s are still in use today. The VT220 omits key #23 (CTRL-J). The PC omits key #16 and key #23.

Each standard macro key is identified by a number, as shown in the following diagrams. These numbers do not appear on your keyboard, but they indicate the order in which each key's macro must appear in the keyboard description. Macro key 1 uses the first macro in the keyboard story, macro key 2 uses the second macro, and so on.

Your system assigns macros to keys based on the order in which the macros appear in the keyboard story. For this reason, your keyboard stories must contain macros for all 24 standard macro keys. Otherwise, your system cannot assign macros correctly. If you do not want to assign a macro to a key, create a macro that does nothing as a placeholder for that key.

While these keys are numbered 1 through 24, there is no 23rd macro key on a VT/DOS keyboard. However, you still must include a 23rd macro in the keyboard story as a placeholder, so that the system interprets the 24th macro properly. CTRL-J is key 23.

VT220 and PC keyboards have their standard macro keys arranged as shown. Each key's number corresponds to the order in which its macro must appear in the keyboard story.

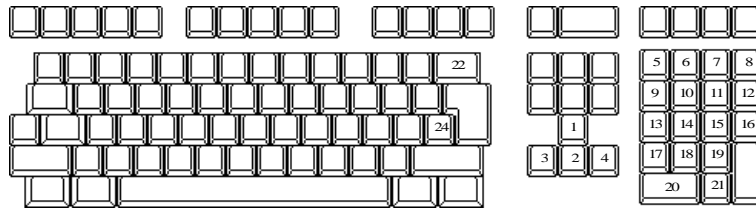


Figure 7-1 Standard Macro Keys (VT220)

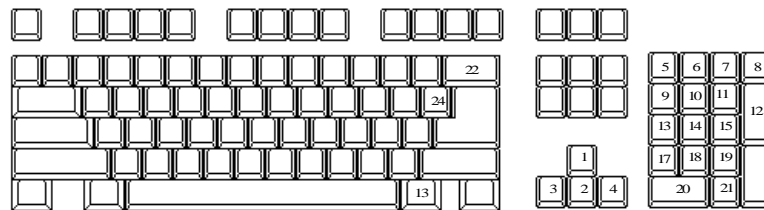


Figure 7-2 Standard Macro Keys (PC)

## Extended Versus Standard Macro Keys

Do not confuse an extended macro key with a standard macro key of the same number. For instance, there is a standard macro key 20 and an extended macro key 20~. While standard macro keys are numbered consecutively 1 through 24 (with the exception of 23), extended macro keys are not. For instance, there are no keys 7~, 8~, 9~, 10~, or 16~.

If your terminal is capable of both VT100 and VT220 operations, you must set it to VT220 7-bit mode to use extended macro keys.

Standard macros are not numbered, and all of them must have definitions to keep them in order. Extended macros are numbered, so unused keys can be omitted.

You do not need to include macros for all the extended macros keys in your story. The only requirement is that you place the extended key macros after the mandatory 24 standard key macros.

## Enabling F13 on the VT220

Many VT220-style terminals are capable of operating as either VT100- or VT220-style terminals. If you have these types of terminals and switch from VT100 to VT220, you alter the use of F13 and its function, because in VT100 mode, F13 generates LF, which the system recognizes as the 23rd macro in the keyboard story. The 23rd macro in a user's keyboard story may contain {insert char}, which causes F13 to function as an Insert Char key. In VT220 mode, this key is extended macro key 25~.

By adding the same macro to extended key 25~, you can cause F13 to execute the same function. This macro would appear somewhere following the mandatory 24 key macros.

Another result of changing to VT220 is that the F12 key no longer functions as a Backspace key. You can program F12 (extended key 24~) to perform the Backspace function. Create a macro similar to the following that moves the cursor left one space and then deletes a character:

```
@24~{cu 1}{del c}
```

## Assigning VT Macros to Extended Macro Keys

Along with the standard 24 macro keys, VT/DOS keyboards have additional extended macro keys, identified by a number followed by a tilde, such as 21~. A VT220 keyboard has 26 of these keys, arranged as in Figure 7-3.

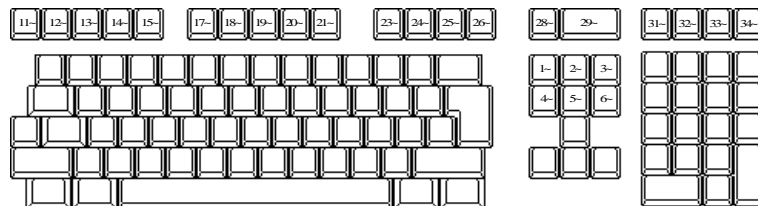
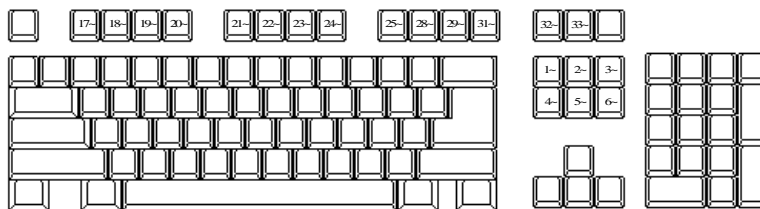


Figure 7-3 Extended Macro Keys (VT220)



The extended macro keys at the upper left of the VT220 keyboard (11~ through 15~) are available only on VT220 and later terminals.

An extended PC keyboard has 20 extended macro keys, arranged as shown in Figure 7-4.



**Figure 7-4 Extended Macro Keys (PC)**



*Do not assign a macro to the Pause/Break key, because your computer uses it for computer recovery. On PC keyboards, the extended macro key 32~ is typically also the PC's print screen key.*

Each extended macro key generates a number followed by a tilde, which your system uses to identify the key and assign it the correct macro. Consequently, the order in which you place extended key macros in a keyboard story does not matter as long as they appear after the 24 standard key macros.

To create an extended key macro, begin the line with an @ in the first column, followed by the number and tilde combination generated by the key to which you want to assign the macro. For instance, F6 generates 17~ on a VT220 keyboard, so the macro for this key would begin with @17~.

Assigning a macro to an extended macro key is the same as for a standard macro key. For instance, you may program the F6 key on a VT220 keyboard (extended programmable key 17~) to take the user to PHONELISTS.STAFF:

```
@17~{go PHONELISTS.STAFF}
```



*An extended macro key is identified by @<number>~. This ~ does not introduce an alternate macro. It is part of the key identifier.*

As with the standard programmable keys, you can assign an alternate macro to an extended programmable key by beginning the alternate macro with a tilde. You may add an alternate macro to F6 that would allow the user to go to WIRES . ADVISORY by pressing Enter (on the numeric keypad) and then F6:

```
@17~{go PHONELISTS . STAFF}~{GO WIRES . ADVISORY}
```

## For VT Users Who Switch Between Keyboards

If someone in your newsroom regularly switches between keyboards, you can make it easier by assigning the `load keyboard` command with the appropriate argument to a programmable key on the user's keyboard. In addition, you can add a macro to the keyboard that the user loads that returns the user to the user's default keyboard.

For instance, you may assign the following `load keyboard` command to F10 on the copy editor's keyboard. Whenever the user needs to use the producer's keyboard, he or she presses F10 (21~):

```
@21~{LOAD KEYBOARD 1}
```

Create a key on the producer's keyboard that allows a user to return to his or her default keyboard description. Assign this command to a programmable key on the producer's keyboard:

```
@21~{LOAD KEYBOARD}}
```

Because this macro executes the `load keyboard` command without naming a keyboard and then presses Enter a second time, it loads the user's default keyboard. Combining two programmable keys makes it easy to move between keyboard description stories.

# CHAPTER 8

## *Forms*

You can use standard Avstar forms that come with the system or customize your database by creating forms and assigning them to queues, based on the kind of information you want to appear in the stories in those queues. You can also use forms in your database to replicate any preprinted forms that you use currently, such as rundowns and assignment sheets.

This chapter explains the following:

- Form Names and Locations
- Guidelines for Designing Forms
- Creating a Form
- Assigning a Form as a Queue or Story Form
- Form Field Types and Definitions
- Standard Avstar Forms
- Mapping Netstation Characters to Avstar

## Form Names and Locations

You create forms, which are stored as stories in queues located in the database file structure under the main system folder, `SYSTEM.FORMS`. The folder name format is `SYSTEM.FORMS.N.NAME`, where *N* is the first letter of the form name. For instance, a rundown form may be `SYSTEM.FORMS.R.RUNDOWN`. You can have up to 255 forms, starting with each letter.

## Guidelines for Designing Forms

Follow these general guidelines in designing forms:

- The story form and queue form can be different, but all fields displayed in the queue form must exist in the story form before you can enter or display data in that field.
- In Avstar you cannot system print wider than 80 columns. If a form extends beyond 80 columns, the extra columns will be dropped from printing.
- A VT interface also truncates beyond 80 columns. If a form displayed on a VT terminal extends beyond 80 columns, the extra columns will not appear on screen.
- If you have a mix of Avstar Workstations and VT terminals, make the label name width of each field one unit greater than the field name width; this will make the fields more readable by providing a one-character separation between them.

## Creating a Form

You can create a new form from any Avstar Workstation. This section outlines the procedures for constructing a basic form for a rundown. After building this form, you can easily modify it to match the rundown you currently use.

The following procedure takes you through steps to create the basic rundown form, which can be used to create any form in the system.

To create a new form at a workstation, do the following:

1. Navigate to the `SYSTEM.FORMS` folder.

The queue for the new form must be stored in the `SYSTEM.FORMS` folder.

2. Click the Tools drop-down menu.

3. Select New Folder.

A new highlighted directory labeled New-Folder appears.

4. Type the name of the folder, such as R.

5. Select the alphabet folder, such as R (the folder for forms with names that begin with the letter R), in which you want to create the new form.

6. Do one of the following:

- a. Click the Tools drop-down menu and select New Queue.

-OR-

- b. Right-click on the folder and select New Queue from the pop-up menu.

A new highlighted file labeled New-Queue appears.

7. Type the new form queue's name, such as Rundown.

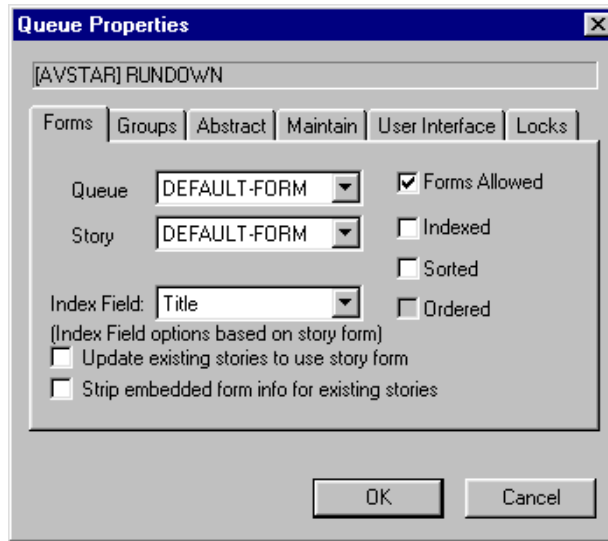
8. Apply the Forms Allowed database trait to the queue by doing the following:

- a. Right-click on the queue in the Directory panel to open the Queue Properties dialog box.

- b. Select the Forms Allowed check box so that a check mark appears.

You can skip steps 3 & 4 if the alphabet folders already exist in the `SYSTEM.FORMS` directory.

- c. Click OK to save changes. Otherwise, you will not be able to



create a form in that queue. See “Changing Database Traits” on page 5-21 and “Forms Tab” on page 5-27 for more information.

9. Double-click the queue to display it in the Queue panel.
10. Do one of the following:
  - a. Click the File drop-down menu and select New Story.
  - OR-
  - b. Position your cursor in the Queue panel and press the Insert key.

If the story form fields do not appear, click on the Story drop-down menu and select the option to Show Form Area.

A new story row appears in the Queue panel and opens in the Story panel. At the top of the Story panel, there is the Story Form panel that displays the story form fields (using the default form or the form previously assigned to the queue). The following graphic

shows the Queue and Story Form panels with some standard form fields.

Slug	Created	Modified	By	P	TIME
	2/22/00 12:29:25	2/22/00 12:29:32	inzr	READY	

Slug	Created	Modified	By	P	TIME
<input type="text"/>	<input type="text" value="2/22/00 12:29:25"/>	<input type="text" value="2/22/00 12:29:32"/>	<input type="text" value="inzr"/>	<input type="text" value=""/>	<input type="text" value=""/>

Queue panel with  
a new story row

Story Form panel  
with six fields

11. Modify each field to customize the new form. For instance, you would need to select each field's type, which defines the field's function within the form. See "Customizing Forms" on page 8-6 and "Form Field Types and Definitions" on page 8-13 for more information.

Since a basic rundown form does not necessarily use the default fields, as shown in the above graphic, field modification is needed. For instance, the form may include the following fields:

- Page field uses the PAGE-NUMBER field type
- Title (or Slug) field uses the TITLE field type
- Presenter (or Anchor) field uses the PRESENTER field type
- Writer (or Reporter) field uses the CREATED-BY field type
- Graphics (or Production Notes) field uses the VAR-N field type, where N is any number
- Audio time field uses the AUDIO-TIME field type
- Back time field uses the BACK-TIME field type

12. Save the story and exit the queue, accepting current fields and form properties as the new form.

## Customizing Forms

After you create the form queue in `SYSTEM.FORMS`, you can modify fields to customize the new form. To customize a form, open its story and do the following:

1. Put the cursor in the Story Form panel and right-click. A pop-up menu will appear.

There are three menu options for customizing form fields: Insert Field, Delete Field, and Field Properties. Access to these options vary, based on whether you right-click on a field in the Story Form panel.

Another option in the pop-up menu is Label Borders. See “Label Borders” on page 8-11 for more information.

2. Choose an option in the pop-up menu, based on one of the following (The choice you make in step 2 also determines which dialog box appears):

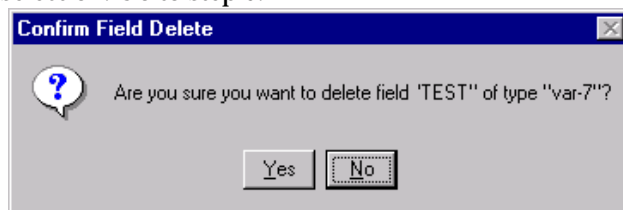
- a. Select Delete Field if you want to remove an existing field from the form. (The field you right-clicked on is the one you will delete if you choose this option.) Figure 8-1 shows the dialog box that appears if you make this selection. Go to step 3.

-OR-

- b. Select Field Properties to modify properties of an existing form field. Figure 8-2 on page 8-7 shows the dialog box that appears if you make this selection. Go to step 4.

-OR-

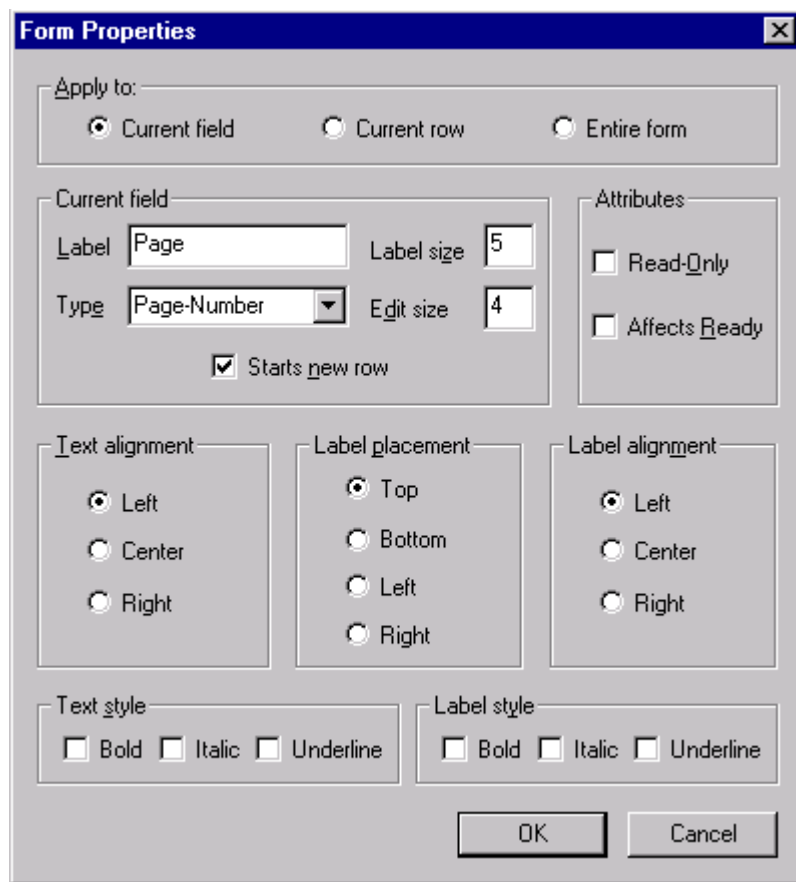
- c. Select Insert Field to add a new field to the form. Figure 8-3 on page 8-8 shows the dialog box that appears if you make this selection. Go to step 5.



**Figure 8-1** Confirm Field Delete Dialog Box



3. Confirm your command to delete the field and, if necessary, return to step 2 to continue modifying form fields.



The image shows a dialog box titled "Form Properties" with a close button (X) in the top right corner. The dialog is organized into several sections:

- Apply to:** Three radio buttons: "Current field" (selected), "Current row", and "Entire form".
- Current field:** A section containing:
  - Label:** A text box containing "Page".
  - Label size:** A text box containing "5".
  - Type:** A dropdown menu showing "Page-Number".
  - Edit size:** A text box containing "4".
  - Starts new row:** A checked checkbox.
- Attributes:** Two unchecked checkboxes: "Read-Only" and "Affects Ready".
- Text alignment:** Three radio buttons: "Left" (selected), "Center", and "Right".
- Label placement:** Five radio buttons: "Top" (selected), "Bottom", "Left", and "Right".
- Label alignment:** Three radio buttons: "Left" (selected), "Center", and "Right".
- Text style:** Three unchecked checkboxes: "Bold", "Italic", and "Underline".
- Label style:** Three unchecked checkboxes: "Bold", "Italic", and "Underline".

At the bottom right, there are two buttons: "OK" and "Cancel".

**Figure 8-2 Form Properties Dialog Box**

4. Use the Apply to radio buttons to determine whether you want your property modification to be applied to:
  - Current field—the one you right-clicked on
  - Current row—the same row of fields as the one you right-clicked on
  - Entire form—all fields in the form

**Insert Field**

Label  Label size

Type  Edit size

Starts new row

Attributes

Read-Only

Affects Ready

Text alignment

Left

Center

Right

Label placement

Top

Bottom

Left

Right

Label alignment

Left

Center

Right

Text style

Bold  Italic  Underline

Label style

Bold  Italic  Underline

Insert Before Insert After Cancel

**Figure 8-3** Insert Field Dialog Box

5. Select various form options to determine appearance and function of field(s) in the form. See Table 8-1 on page 8-9 for an explanation of the form options.

**Table 8-1 Form Options**

Option	Explanation
Label	Enter current field's name you want to appear in the Story Form panel.
Type	Select a field type from the drop-down list, which defines the field's function. See "Form Field Types and Definitions" on page 8-13 for a detailed explanation of the various types you can choose, including variable fields that allow you to make up your own field names, such as "shot" or "printed."
Starts new row	Select this check box to force the field to be the first one in the next row of the form.
Label size	<p>Enter a numerical value to determine the label size.</p> <p>This size is in approximate characters. Using a proportional font will, of course, cause the number of characters to vary.</p> <p>Generally, label size should be set to Edit size plus one.</p>
Edit size	<p>Enter a numerical value to determine the space text will fill in the field.</p> <p>System administrators can set a limit for text in Story Form fields, using a Registry value defined as <code>VT Compatibility</code> at each workstation. See "VT Compatibility" on page F-18 for more information.</p>
Attributes	<p>Select attributes for the form field.</p> <ul style="list-style-type: none"> <li>• Read-Only check box determines whether the form field can be read (not modified).</li> <li>• Affects Ready check box determines whether the form field participates in determining the Ready field value.</li> </ul>
Text alignment	Select text position within the field.

**Table 8-1 Form Options**

<b>Option</b>	<b>Explanation</b>
Label placement	Select label position. <ul style="list-style-type: none"><li>• Top (default) puts label on top of the field</li><li>• Bottom puts label below the field</li><li>• Left or Right puts label to either side of the field.</li></ul>
Label alignment	Select way in which label aligns (left, center, or right) with the field.
Text style	Select appearance of text (bold, italic, or underline) in the field.
Label style	Select appearance of the label (bold, italic, or underline).

6. Do one of the following:

a. Click OK in the Form Properties dialog box to record changes and close the dialog box.

-OR-

b. Click Insert Before or Insert After in the Insert Field dialog box, depending on where you want the new field to appear in the form in relation to the field (or cursor position) you right-clicked on in step 1. This will record the changes and close the dialog box.

7. Save story in the form queue, by using Save button or Save option in the File drop-down menu.

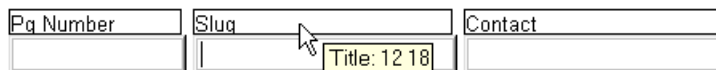
Once a form is created and customized, it can be assigned to other queues in the database as either the queue form, which dictates appearance of the queue, or story form, which dictates appearance of stories created in the queue.

## Label Borders

Label borders provide various information about fields in a tool tip format. To turn on Label borders in the Story Form panel, do the following:

1. Right-click in the Story Form panel.
2. Select Label Borders in the pop-up menu.
3. Position mouse pointer over a field label to view tool tip.

When selected, Label borders place rectangular borders around the labels for each field in the form. When the mouse pointer is positioned over a field's label, a tool tip appears that displays that field's type, and the character count for the field position and size.



For instance, the field called Slug (shown above) is a TITLE field located 12 characters in from the left with a Label size of 18 characters. So, from the position of the Slug field, you can determine the Pg Number field has a Label size of 12 characters. Also, the Contact field is located 30 characters in from the left—that is, its position in the row is equal to the sum of the Label sizes for the fields directly to its left. You could confirm these figures by positioning your mouse over the fields and viewing the tool tips that appear.

## Assigning a Form as a Queue or Story Form

Queue and Story forms are database traits that allow you to assign different forms to different folders and queues using the Directory/Queue Properties dialog box at the workstation, or the `dbtraits` command at the console. For more information on how to assign a form at the console, see “Queue Form” and “Story Form” on page G-23.

You assign these traits to define the appearance of information in the Queue panel and Story Form panel.

To assign a form at an Avstar Workstation, do the following:

1. Navigate to the directory (folder) or queue you want in the Directory panel.
2. Right-click to open the Directory/Queue Properties dialog box. Access to this dialog box and its appearance varies, depending on certain circumstances. See “Directory/Queue Properties Dialog Box” on page 5-25 for more information.

The screenshot shows the 'Directory Properties' dialog box with the 'Forms' tab selected. The title bar reads 'Directory Properties' and the path is '[AVSTAR] SHOWS'. The 'Forms' tab is active, with other tabs being 'Groups', 'Abstract', 'Maintain', and 'User Interface'. The 'Queue' dropdown is set to 'RUNDOWN-FORM' and the 'Story' dropdown is also set to 'RUNDOWN-FORM'. The 'Index Field' dropdown is set to 'TITLE'. There are three checkboxes: 'Forms Allowed' (unchecked), 'Indexed' (unchecked), and 'Sorted' (unchecked). Below these are two more checkboxes: 'Update existing stories to use story form' (unchecked) and 'Strip embedded form info for existing stories' (unchecked). At the bottom, there is a checkbox for 'Apply changes to all subdirectories and queues' (unchecked), and 'OK' and 'Cancel' buttons.

3. Do either or both of the following:
  - a. Use the Queue drop-down list on the Forms tab to select the form you want to apply to the directory as queue form database trait.
  - b. Use the Story drop-down list on the Forms tab to select the form you want to apply to the directory as story form database trait.

See “Forms Tab” on page 5-27 for more information.

4. Click OK to save changes and apply the new queue/story form settings.



*Users should log off and sign back on to view the new queue/story form settings.*

## Form Field Types and Definitions

The Avstar form field types are explained in Table 8-2. Included is the suggested maximum or minimum length for each field (where applicable) and whether a user can enter text in the field.

You can repeat only the variable and “affects status” fields in a form. You can use all other fields only once.

Following Table 8-2 is an explanation of some forms pertaining to broadcast and/or machine control and which fields are typically used in a variety of forms/queues.

**Table 8-2 Avstar Form Fields**


Field Type	Description
AFF-READY-N (N represent any number)	AFF-READY-N fields are created when a system is converted from Avid Netstation to Avstar. The AFF-READY-N fields are assigned the “Affects Ready” attribute when created. This means the field affects the display in the READY status field and allows a user to initialize a story and change its status. There can be more than one AFF-READY-N field in a form. If any AFF-READY-N field within a form contains a “?” then the READY field displays a NOT READY message. If a question mark does not appear in any of the AFF-READY-N fields in the form, then the Ready field displays a READY message. This field should precede the READY status field in the form. In Avstar, the “Affects Ready” attribute can be assigned to any field in a form and the result would be the same behavior as described here pertaining to the AFF-READY-N field. For more information, see the definition for the READY field in this document.

**Table 8-2 Avstar Form Fields (Continued)**

Field Type (Continued)	Description (Continued)
AIR-DATE	<p>When a story is aired using the show-timing function, the date and time it airs is inserted in this field. This field is designed to show which story is currently on-air and give users a sense of how much time remains until a later story goes to air. As the producer syncs timing on stories, the queue display on other Avstar Workstations shows that story in a different color. (Peach is the standard default color rule.) The format for the date and time on the VT interface is Sat Apr 23 12:01 1998. The format on Avstar Workstation is controlled by the workstation control panel for regional settings. When used for VT sessions, the format is sensitive to the LOCALE setting which controls localization of the system. It consists of two components: locale's appropriate date representation" and "locale's appropriate time representation." If the program finds a definition for "VT_TIME," then it will format the dates according to the UNIX "strftime" man page. On the VT interface, if the field length is less than 21 characters, the time is truncated in the same way as in the modified time field. Widths of 12, 16, and 21 characters work best. You cannot enter data in this field.</p>
<p>APPN-X to APPN-X (N represents 1 through 5; X represents any number)</p>	<p>These fields are used differently, depending on the queue in which they are used. They appear in Machine Control Terminal forms and some Broadcast Control System (BCS) forms. Also, many of the fields have different meaning, depending on the device type. For instance, in an MCT form, the event ID for a video/cart machine goes in the APP1-1 field, but in the APP2-1 field for a CG or SS, where the APP1-1 field is used for the style. These fields were previously used in the mail and account forms in Avid Netstation, but are no longer used for those forms in Avstar NRCS. The APPN-X fields have been renamed in Avstar and occur only in template conversion from Avid Netstation.</p>



**Table 8-2 Avstar Form Fields (Continued)**

Field Type (Continued)	Description (Continued)
AUDIO-TIME	<p>This field will display the estimated time for reading a story, which can be estimated by the computer system or entered into the field by a user. If a user enters a time in the field, the system will use it to calculate the total time. If there is also a tape-time field in the form, the system adds the TAPE-TIME to the AUDIO-TIME to calculate the story's total time. Without any user input, the system will display an estimated time based on the length of the story and presenter's read rate, which is obtained from the PRESENTER field in the form. If there is no PRESENTER field or it does not contain a user ID, then audio time is based on the system's default read rate. The length of the story is actually the word count of the story, since the read rate is based on words per minute. If a user has entered a time in the field and wants to restore the audio time calculated by the system, the user should remove entered data from the field with the space bar, delete, or backspace key. After the cursor leaves the form field, the system will then display the computer-calculated audio time and recalculate total time accordingly.</p>
	<p><i>The AUDIO-TIME and TAPE-TIME fields should not exceed nine characters in length. This only applies to mixed-client environments with Avstar Workstations and VTs connected to PCUs. The typical size of each of these fields is no more than five characters, such as 00:00</i></p>
BACK-TIME	<p>The system displays the back-time in this field. The back-time field is usually eight characters wide, such as 00:00:00. A user can enter data in this field to indicate hard-hit times for back-timing to certain points within a program.</p>
CA-CAPTURED	<p>This field displays total number of characters captured during a session connection; it is one of eight special fields used in the SYSTEM.ACCOUNT queue form for logging connection time activity. Although this field is not required, omitting it will prevent Avstar from displaying the corresponding information in the form.</p>
CA-DIRECTION	<p>This field displays the direction of incoming or outgoing connections. It is one of eight special fields used in the SYSTEM.ACCOUNT queue form for logging connection time activity. Although this field is not required, omitting it will prevent Avstar from displaying the corresponding information in the form.</p>

**Table 8-2 Avstar Form Fields (Continued)**

<b>Field Type (Continued)</b>	<b>Description (Continued)</b>
CA-ELAPSED	This field displays elapsed time of a session connection. It is one of eight fields used in the <code>SYSTEM.ACCOUNT</code> queue for logging connection time activity. Although this field is not required, omitting it will prevent Avstar from displaying corresponding information in the form.
CA-IDENT	This field displays the connection identifier. It is one of eight fields used in the <code>SYSTEM.ACCOUNT</code> queue for logging connection time activity. Although this field is not required, omitting it will prevent Avstar from displaying corresponding information in the form.
CA-ORIGIN	This field displays the origin machine name. It is one of eight special fields used in the <code>SYSTEM.ACCOUNT</code> queue form for logging connection time activity. Although this field is not required, omitting it will prevent Avstar from displaying corresponding information in the form.
CA-RECEIVED	This field displays the total number of characters received from a remote system during a connection. It is one of eight special fields used in the <code>SYSTEM.ACCOUNT</code> queue form for logging connection time activity. Although this field is not required, omitting it will prevent Avstar from displaying corresponding information in the form.
CA-REMOTE	This field displays the remotely connected machine name. It is one of eight special fields used in the <code>SYSTEM.ACCOUNT</code> queue form for logging connection time activity. Although this field is not required, omitting it will prevent Avstar from displaying corresponding information in the form.
CA-SENT	This field displays the total number of characters sent to a remote system during a connection. It is one of eight special fields used in the <code>SYSTEM.ACCOUNT</code> queue form for logging connection time activity. Although this field is not required, omitting it will prevent Avstar from displaying corresponding information in the form.

**Table 8-2 Avstar Form Fields (Continued)**

Field Type (Continued)	Description (Continued)
CG-ID	<p>This field holds the ID's or recorded page addresses from the character generator on which a super is written by the CG interface. In some cases, particularly when the field is used as the "sortfield" for a queue and all other database issues, this field is referred to as CG-ADDR. For instance, the command line for assigning this field as the sortfield in the database would be: <b>dbtraits &lt;queue name&gt; sortfield cg-addr</b></p> <p>This field is primarily used in association with Machine Control Systems and Broadcast Control Systems. Please see the "Avstar MCS/BCS Fields and Forms" on page 8-24 for further information on when and how this field is used.</p>
CG-TEMPLATE	<p>This field contains template information for the character generator, namely the address on the character generator of the template or tab field to be used for the requested super. This field is primarily used in association with Machine Control Systems and Broadcast Control Systems. Please see the "Avstar MCS/BCS Fields and Forms" on page 8-24 for further information on when and how this field is used.</p>
CG-TEXT	<p>This field contains text of the super from the machine control instruction requested by a user in the script. It is written into specified template fields on the character generator interfaced with Avstar NRCS. This field is primarily used in association with Machine Control Systems and Broadcast Control Systems. Please see the "Avstar MCS/BCS Fields and Forms" on page 8-24 for further information on when and how this field is used.</p>
CHANNEL	<p>The letter or numerical identifier of the on-air output or playback channel of an interfaced production device is located in this field. Most production devices have two or more channels. This field is primarily used in association with Machine Control Systems and Broadcast Control Systems. Please see the "Avstar MCS/BCS Fields and Forms" on page 8-24 for further information on when and how this field is used.</p>
CREATE-BY	<p>When you open a new story, the system enters your user name in this field. The name is permanent and cannot be erased or overwritten. In a Mail form, this field is used to indicate who sent the E-mail.</p>

**Table 8-2 Avstar Form Fields (Continued)**

Field Type (Continued)	Description (Continued)
CREATE-DATE	When you create a new story, the system stores the date and time the story was created in this field. The format for the date and time on the VT interface is Sat Oct 21 12:01 2000. The format on Avstar NRCS is controlled by the workstation control panel for regional settings. On the VT interface, if the field length is less than 21 characters, the time is truncated, as in the modified time field. Widths of 12, 16, and 21 characters work best. You cannot enter data.
CUME-TIME	The computer displays the cumulative (cume) time. This field represents the time of all the stories—except the selected story—added together. It has a suggested minimum length of five characters, and is typically eight characters.
DM-NAME	This field, identified as DEVICE- MGR when stored in the database, displays the device name controlling a particular event. This field is primarily used in association with Machine Control Systems and Broadcast Control Systems. See the “Avstar MCS/BCS Fields and Forms” on page 8-24 for further information on when and how this field is used.
ENDORSE-BY	This field enables a user with write-access to the queue to endorse stories in that queue. When a story is first saved in a queue, the field is red and blank. Whenever a user endorses the story, the system places that user’s name in the ENDORSE-BY field corresponding to that story and the field changes to green. To manually endorse a story, click on the field in the queue. If another user subsequently changes the story and saves it, the ENDORSE-BY field turns yellow but the endorser’s name remains in the field. This indicates the story was changed by one user after it had been approved by another. The endorser can see who made the change by looking in the MODIFY-BY field if there is one in the story form. The endorser can also withdraw approval by opening the story form and deleting the user id from the ENDORSE-BY field. If the ENDORSE-BY field is not shown in the Queue panel, a user can still endorse the story by typing a character in the ENDORSE-BY field located in the story form. A non-system administrator (non-superuser) cannot kill an endorsed story if another user has the Production Lock set.



*The story form must include a MODIFY-BY field to show the green "endorsed" status in the ENDORSE-BY field.*

**Table 8-2 Avstar Form Fields (Continued)**

Field Type (Continued)	Description (Continued)
EVENT-DURATION	This field, identified as DURATION when stored in the database, displays the playing length of an event, such as how long a certain piece of video will be aired or the duration of a CG or still store effect. This field is primarily used in association with Machine Control Systems and Broadcast Control Systems. See the “Avstar MCS/BCS Fields and Forms” on page 8-24 for further information on when and how this field is used.
EVENT-EFFECT	This field, identified as EFFECT when stored in the database, holds the effect name requested in association with a machine control or broadcast control event that will be applied to the character generator or still store machine when it’s taken to air. For instance: a wipe or a dissolve. This field is primarily used in association with Machine Control Systems and Broadcast Control Systems. See the “Avstar MCS/BCS Fields and Forms” on page 8-24 for further information on when and how this field is used.
EVENT-STATUS	This field displays availability and play status of an MCS/BCS event, as reported by the production device involved. For instance, a video event could be reported as N/L (not loaded), CUED,PLAYING, or STOPPED, among other things. In rundown and Event List forms only the status of a video event can be displayed. In Machine Control Terminal and Broadcast Control Workstation forms, this field can also contain the status of CG and still store events. EVENT-STATUS is primarily used in association with Machine Control Systems and Broadcast Control Systems. See the “Avstar MCS/BCS Fields and Forms” on page 8-24 for further information on when and how this field is used.
EVENT-STYLE	This field, identified as STYLE when stored in the database, contains the MCS/BCS style name specified when a user requests a CG or Still Store event in the production cue. It is typically an alpha or alphanumeric sequence that is a maximum of eight characters long. For instance, Avstar NRCS translates a CG style into an address on the character generator at which a template is stored. That template is then used to build the requested super. Styles are defined in stories in the SYSTEM.RESOURCE queue. They define the details, such as CG template, number of fields, still preset or playback effect, that define an event. This field is primarily used in association with Machine Control Systems and Broadcast Control Systems. See the “Avstar MCS/BCS Fields and Forms” on page 8-24 for further information on when and how this field is used.

**Table 8-2 Avstar Form Fields (Continued)**

Field Type (Continued)	Description (Continued)
LINE-COUNT	This field is used in queues for wire services. It displays the number of lines belonging to each story or document in the queue. It cannot be edited by the user. It is strictly an informational field to assist the user in estimating story length from the queue. The LINE-COUNT field is only relevant to VT sessions. Avstar NRCS does not maintain or update this field since the display of the data would vary greatly depending on window size used on each Avstar Workstation. Additionally, any stories carried over to Avstar NRCS from a Basys system will have an empty LINE-COUNT field initially or until re-edited from a VT session.
LITERAL	This field is a non-editable field, typically used as a label or spacer to assist in alignment of other fields within the Story Form.
MAIL-CC	This field is used in the Mail story form to display names of users receiving a copy of an E-mail message.
MAIL-TO	This field is used in the Mail story form to display names of users to whom an E-mail message is sent.
MEDIA-ID	This field displays the Media ID, a unique identifier used to access a machine control event on a production device. This field is exclusively used in association with the Broadcast Control Workstation (BCWS) form. It is a collective category that contains values of three other fields: VIDEO-ID, CG-ID, and STILL-ID, allowing them to be combined in a single column. See the "Avstar MCS/BCS Fields and Forms" on page 8-24 for further information on when and how this field is used.
MODIFY-BY	This field contains the name of whoever most recently modified a story. It can be set to hold less than eight characters; however, if it is set to less than eight, the users' names are truncated. A user cannot edit this field.
MODIFY-DATE	Displays date and time story was last modified. Every time a story is edited and saved, the system updates this field.
MODIFY-DEV	If a device name for a workstation is included in the configuration file, it appears in this field when a story is saved at that workstation. If device name is not included in the configuration file, this field remains blank. The maximum number of characters is eight. You cannot enter data in this field.

**Table 8-2 Avstar Form Fields (Continued)**

Field Type (Continued)	Description (Continued)
NSML-LITERAL	This field is only seen in stories and forms transferred from an Avid Netstation database to an Avstar NRCS database and cannot be created by a user. The field is a by-product of the database conversion process and represents a protected field in the Netstation template.
PAGE-NUMBER	This field is used primarily to arrange stories in a queue. If a queue is set to sort stories by page number, users can change the order of stories in the queue by entering in new page numbers. This is commonly done with run-downs, because it allows the producer to quickly change the order of stories by assigning them new page numbers. This field can also be displayed on a serial prompter or the WinCue prompter. The system uses the first six characters of the field as a page number when the story is printed. This field is used in printing only when the story is printed using the “print script” option. The suggested length for this field is six characters.
PRESENTER	The system uses the name of the user from this field to look up the user read-rate. The system then calculates audio time for the story based on the user read-rate. If a form does not contain a PRESENTER field, the system calculates the audio time, based on the default read-rate set in the system profile. (This is usually 180 wpm in English.)
READY	In some cases, the system places status information, such as NEW, HOLD, LOCKED, or WIRE in this field. For instance, a wire story starts with the WIRE status. When you change the story, it changes from WIRE to READY. In the (VT) Mail form, the READY field is used to indicate whether an E-mail message is NEW or was READ. This field is six characters. You cannot enter data in this field, but it can be altered based on information put in other fields with the “Affects Ready” attribute in the form. In these cases, Avstar checks the “Affects Ready” fields in the form for a “?” and if any one of the fields are empty or contain that character as the first non-blank character, then the READY field displays a NOT-READY status. However, Avstar displays READY in the READY field if none of the fields with the “Affect Ready” attribute contain a question mark.
RESULT-INDEX	This field is in forms used to define the display of Avstar database search results. It contains the sequence number that indicates the original order of items within the search results.

**Table 8-2 Avstar Form Fields (Continued)**

Field Type (Continued)	Description (Continued)
RESULT-LOCATION	This field is in forms used to define the display of Avstar database search results. It contains the location of data found during a search, such as the name of a queue containing the story found matching the search criteria.
SEARCH-ID	This field, which contains the request ID number, is used by the Find All and Fast Text Search features of Avstar.
STATUS	This field is used in a rundown queue. It will display “OK” or “ERROR,” depending on the machine control event. The Broadcast Control System monitoring program sets this field to indicate whether there are any errors in the production cues in stories of the rundown. In the <code>SYSTEM.ACCOUNT</code> queue form used for logging connection activity, the STATUS field is used to display the type of connection.
STILL-ID	In an Event List or Composite List queue, this field displays the alphanumeric identifier for a still store graphic. This field is primarily used in association with Machine Control Systems and Broadcast Control Systems. See the “Avstar MCS/BCS Fields & Forms” section of this document for further information on when and how this field is used.
STILL-PRESET	This field contains the number or letter designation of a predefined still store format. It is typically used in the form for the still store device event list, and is recognized by Avstar’s Broadcast Control Workstation. This field is primarily used in association with Machine Control Systems and Broadcast Control Systems. See the “Avstar MCS/BCS Fields and Forms” on page 8-24 for further information on when and how this field is used.
TAPE-TIME	A user can enter the tape run-time in this field. If there is an audio-time field, the system adds tape time to audio time to calculate the story’s total time. The minimum length is four characters. When located in the rundown/MCT forms, the TAPE-TIME field can optionally be set to the actual video duration by the Machine Control System device managers for the Quantel Clipbox, Sony Betacart, and Avid AirPlay.



*The AUDIO-TIME and TAPE-TIME fields should not exceed nine characters in length. This only applies to mixed-client environments with Avstar Workstations and VTs connected to PCUs. The typical size of each of these fields is no more than five characters, such as 00:00*



**Table 8-2 Avstar Form Fields (Continued)**

Field Type (Continued)	Description (Continued)
TITLE	<p>This field is used to give each story a name. Occasionally referred to as the “Slug” field, it is the default field for sorting and indexing. The suggested length for this field is no more than 20 characters. The TITLE field is usually selected as the “index field.” That means it is the field searched when a user conducts a Find or Find All search function and specifies a search of the index field. In a Mail form, this field is used to display the Subject of the E-mail message sent. In the <code>SYSTEM.ACCOUNT</code> queue form used for logging connection activity, the TITLE field is used to display the Service name.</p>
TOTAL-TIME	<p>The computer stores the total time for a story in this field, calculated based on the sum of information from the TAPE-TIME and AUDIO-TIME fields. Tape and audio times are added to determine the total story time when calculating back-time. If only audio-time is present, that field is used as the total time. The TOTAL-TIME field has a minimum of four characters. This field is supplied by the system.</p>
VAR-N (N represents any number.)	<p>This variable field was carried over from the Avid Netstation template conversion. It is typically used for generic editable text fields. Users can make up their own new fields using descriptive names, such as “crew” for a field that lists a reporter/photographer team assigned to cover a story. The user can employ anything for a field name with the following restrictions:</p> <ul style="list-style-type: none"> <li>• The name must be 12 characters or less</li> <li>• The name must begin with a letter of the alphabet</li> <li>• It can include any letters (a-z), numbers (0-9), a dash (-) or a period (.)</li> <li>• The name is not case sensitive. User-created field names are treated the same as variable fields.</li> </ul>
VIDEO-ID	<p>This field is used in the rundown queue to display the tape number or clip ID for video. It is also found in forms for the composite and video event lists. This field is primarily used in association with Machine Control Systems and Broadcast Control Systems. See the “Avstar MCS/BCS Fields and Forms” on page 8-24 for further information on when and how this field is used.</p>

**Table 8-2 Avstar Form Fields (Continued)**

Field Type (Continued)	Description (Continued)
WRITER	When someone creates a new story, the system automatically enters his or her user name in this field if it is present in the form/queue. You can also enter text and change the name. This field should be at least eight characters. However, if the field is used in a wire form, it should be 10 characters, so the story's wire source name fits in the field.

## Avstar MCS/BCS Fields and Forms

Certain field options within Avstar are used in specialized forms, which deal with various devices that interface with the Avstar Newsroom Computer System. These forms work in part with the Machine Control System (MCS) or Broadcast Control System (BCS) monitoring programs.



*The application fields available in Avid Netstation are now used only by the Machine Control Terminal, which cannot use new field types because of the limitation in the number of fields usable by CCU/PCU programs.*

In certain cases, some fields in these forms are filled out by the system rather than users. The BCS Monitor program in Avstar NRCS recognizes a new set of descriptive MCS field types in the Rundown, Event list, and Composite list forms.

Here's a brief description of these forms:

Data Type	Rundown Forms
RUNDOWN FORMS	This is the form used by stories in the rundown queue. The monitor program may extract text from some fields in this form and may put text in others. It also copies text from the rundown form to Event List/Composite List and MCT forms. Only event information for video devices is currently placed in the rundown story form.

Data Type	Rundown Forms
EVENT LIST FORMS	The Monitor program places information about an event destined for a specific device in the form fields of the Event List story. It also copies rundown field text—for instance, title and page number—into matching fields in the Event List story form.
COMPOSITE LIST FORMS	The Monitor program puts information about events destined for all devices in the form fields of the Composite List story. It also copies rundown field text—for instance, the title and page number—into matching fields in the Event List story form.
MCT FORMS	All MCS device "drivers" use a Machine Control Terminal form to hold displayable data for the MCT monitoring program. Some event status is passed through the fields in this form.
BCWS FORMS	The function of the Broadcast Control Workstation form is similar to the MCT form in that it controls which data will be displayed on the playback screen. Currently, the BCWS only extracts the PAGE-NUMBER from this form, but in future releases it will use all fields in the form. Fields defined for use only by the BCWS are: MEDIA-ID, EVENT-DURATION (also known as DURATION), CHANNEL, and DM-NAME (also known as DEVICE-MGR).

Below is a chart of form fields filled in by the Monitor programs.

<b>Data Type</b>	<b>Rundown Forms</b>	<b>Event List Forms</b>	<b>Composite List Forms</b>	<b>MCT Forms (MCS only)</b>
<b>Error Status</b>	STATUS			
<b>Video ID/ Address</b>	VIDEO-ID	VIDEO-ID	VIDEO-ID	APP1-1
<b>Video Status</b>	EVENT-STATUS	EVENT-STATUS	EVENT-STATUS	AFF-READY-1
<b>Video Duration</b>	TAPE-TIME			TAPE-TIME
<b>CG Address</b>		CG-ID (CG-ADDR)	CG-ID (CG-ADDR)	APP2-1
<b>CG Template</b>		CG- TEMPLATE	CG- TEMPLATE	APP3-1
<b>CG Text</b>		CG-TEXT	CG-TEXT	VAR-1
<b>Still Store Address</b>		STILL-ID	STILL-ID	APP2-1
<b>Still Store Preset</b>		STILL- PRESET	STILL- PRESET	APP3-1
<b>CG/SS Style</b>		EVENT- STYLE	EVENT- STYLE	APP1-1
<b>CG/SS Effect</b>		EVENT- EFFECT	EVENT- EFFECT	MODIFY-DEV
<b>CG/SS Event Status</b>				AFF-READY-1
<b>MCT Current Event</b>				WRITER

<b>Data Type</b>	<b>Rundown Forms</b>	<b>Event List Forms</b>	<b>Composite List Forms</b>	<b>MCT Forms (MCS only)</b>
<b>Sony Barcode SOM</b>	APP4-1	APP4-1		
<b>Sony Barcode EOM</b>	APP5-1	APP5-1		

## Standard Avstar Forms

The Avstar system contains a number of standard forms, each designed for use with one of the system's features. The forms your system provides are ready to use, but you can also customize them. You can add or remove fields from a form or change a field's title.

### Account Queue Form

The system keeps a record of events, such as who logged into a connect session, the service used, and length of the connect session. This information is made available when the system creates a story in the `SYSTEM.ACCOUNT` queue and puts information in the story's form.

A form designed for this must be assigned to `SYSTEM.ACCOUNT`, or your system cannot display the information. The standard account queue form is usually stored in the queue `SYSTEM.FORMS.A.ACCOUNT`.

Table 8-3 shows the fields used in this form.

**Table 8-3 Account Queue Form Fields**

<b>Field Label</b>	<b>Field Type</b>
Service name	TITLE
User	CREATE-BY
Date and time	CREATE-DATE
Connection type	STATUS
Origin machine name	CA-ORIGIN
Incoming or outgoing connection	CA-DIRECTION
Remote machine name	CA-REMOTE
Elapsed time of session connection	CA-ELAPSED
Connection identifier	CA-IDENT
Total characters sent to remote system	CA-SENT
Total characters received from remote system	CA-RECEIVED
Total characters captured	CA-CAPTURED

You are not required to include any of these fields in the account queue form. However, omitting a field prevents the system from displaying information associated with that field in the account queue.

## Mail Form

The Mail feature has a form specially designed for mail stories. This form has fields for elements such as the subject of your message, name of message recipient, and people you want copied on the message. The mail form is stored in `SYSTEM.FORMS.M.MAIL`. The system includes a mail form, but you can specify a form for the mail feature to use by assigning the form to the `SYSTEM.MAIL.OUT` queue.

You may modify this form to suit your system needs. The following fields are used by the Mail queue.

**Table 8-4 Mail Form Fields**

Field Label	Field Type
To	MAIL-TO
Subject	TITLE
Cc	MAIL-CC
From	CREATE-BY

Only the To field is required. A form that does not have the other fields is still usable.

## Timing Form

When you back-time a show while it is on the air, your system displays a timing form. This form contains fields the system uses to display the show's timing parameters. This form is used with VT interfaces only.

The system requires that certain fields exist in this form to display timing information.

Table 8-5 shows field types used in the timing form.

**Table 8-5**      **Timing Form Fields**

<b>Field Label</b>	<b>Field Type</b>
Current Time	TAPE-TIME
Show Length	TOTAL-TIME
On-Air	CREATE-BY
Off-Air	MODIFY-BY
Over/Under	AUDIO-TIME
Elapsed	CUME-TIME
Remaining	BACK-TIME

None of these fields is required. However, the system uses the fields to calculate information regarding the show's timing; omitting any of them prevents display of that information.

## Print Form

The system maintains a separate print queue for each of your printers. When someone sends a story to a printer, the system places it in the printer's print queue behind any pending print requests.

While a print request is held in a print queue, the system displays information in the print request's form, such as creator of the print request and time the request was made. To display this information, the print queue must have a form designed to display print queue information.

Your system's standard print queue form is assigned to your system's print queues. This form is stored in `SYSTEMS.FORMS.P.PRINTER`.



Table 8-6 shows field types used in the print queue form.

**Table 8-6 Print Form Fields**

<b>Field Label</b>	<b>Field Type</b>
Title/Folder	TITLE
Date Printed	CREATE-DATE
By	CREATE-BY
Style	VAR-1
Type	STATUS
Copies	PAGE-NUMBER

None of these fields is required. However, omitting any field would prevent the display of information associated with that field.

## Seek Form

This form is used by someone using the VT `seek` command on a Find All operation on Avstar Workstation.

When someone enters the `seek` command to conduct a background search, your system responds by presenting the user with a seek form. This form contains a number of fields the user must fill in to specify how to perform the search. You can use this form as it is or modify it.

Your system uses the form assigned to `SYSTEM.SEEK`, or whichever queue is specified for `seek` in system's dictionary (`/site/dict/`). Since the system expects to find certain information in each of the form's fields, the form assigned to this queue must be designed for use with the `seek` command.

Table 8-7 shows field types used in the Seek form.

**Table 8-7      Seek Form Fields**

<b>Field Label</b>	<b>Field Type</b>
Id	SEARCH-ID
Status	READY
Search For	VAR-1
Search Where	APP-1
Results Queue	TITLE
Search Type	APP2-1
Max Hits	PAGE-NUMBER
Notify Hits	APP3-1
Hits so Far	PRESENTER
By	CREATED BY
Started At	CREATE-DATE



*The seek command must have certain information to perform a search. If you leave out a required field, the seek server displays a `Missing form field` message whenever someone tries to use the seek command.*

## Wire Story Form

To display information such as title, time moved, source, and average reading time in each wire, the system contains a wire story form. It is assigned to the wire queues and stored in `SYSTEM.FORMS.W.WIRE`.

Table 8-8 shows field types used in the wire story form.

**Table 8-8 Wire Story Form Fields**

<b>Field Label</b>	<b>Field Type</b>
Title	TITLE
From	WRITER or CREATE-BY
Moved	CREATE-DATE
Status	READY
Time	AUDIO-TIME
FlashWord	PRESENTER
Number	PAGE-NUMBER
Lines	LINE-COUNT

Any changes you make to this form do not take effect until you restart the wire.

## Mapping Netstation Characters to Avstar

Table 8-9 lists Avid Netstation template characters mapped to Avstar form field names. You will need this information when you:

- Convert stories from the Netstation system to Avstar NRCS
- Use tx/net to transmit a story from Avstar NRCS to Netstation or Archive II systems

**Table 8-9 Netstation Characters Mapped to Avstar Fields**

Netstation Character	Netstation Definition	Avstar Field Name
X	Affects status	AFF-READY-N
A	Anchor	PRESENTER
1-5	Application fields	APP1- N to APP5-N
a	Audio time	AUDIO-TIME
q	Back time	BACK-TIME
c	Created by	CREATE-BY
Q	Cume time	CUME-TIME
d	Devname	MODIFY-DEV
E	Endorsed by	ENDORSE-BY
l	Line count	LINE-COUNT
M	Modified by	MODIFY-BY
m	Modified time	MODIFY-DATE
o	Originate time	CREATE-DATE
p	Page number	PAGE-NUMBER

**Table 8-9 Netstation Characters Mapped to Avstar Fields**

<b>Netstation Character</b>	<b>Netstation Definition</b>	<b>Avstar Field Name</b>
P	Protected	<i>obsolete</i>
S	Slug	TITLE
x	Status	STATUS
i	Tab line	<i>obsolete</i>
t	Tape time	TAPE-TIME
T	Total time	TOTAL-TIME
V	Variable	VAR-N
W	Writer	WRITER



# SECTION II

## *System Setup and Configuration*

The section contains the following chapters:

- Chapter 9, *Character Generator Title Entry*
- Chapter 10, *Ed, the Line Editor*
- Chapter 11, *Configuration Files*
- Chapter 12, *Printers*
- Chapter 13, *Wires*
- Chapter 14, *Servers*
- Chapter 15, *Web Publishing*
- Chapter 16, *Web Access*

---



# CHAPTER 9

## *Character Generator Title Entry*

The Avstar Newsroom Computer System (NRCS) provides journalists, producers, directors, writers, and technical personnel in a newsroom with an array of tools to make their jobs easier. One such tool is the CG Title Entry feature, which enables newsroom personnel to simulate character-generated graphics at the Avstar Workstation. By offering a graphical view of CG information, Avstar NRCS helps production staff verify the accuracy and quality of the data prior to air.

This chapter contains:

- Overview of CG Title Entry
- Title Entry Setup and Configuration
- CG Template Backgrounds
  - Required Bitmaps
  - Capture Tool
- CG Template Editor
- Title Entry Security

## Overview of CG Title Entry

When writing scripts, journalists are typically required to include information in the script that will appear as a character-generated graphic when the story is aired. These graphics, called supers or CGs, must be entered into the script in such a way that Avstar NRCS can identify the information as data for the character generator. This method of identification uses production cues—also called machine instructions—which are separated from the story text. Here is an example of a production cue for a CG with two lines of text for the graphic and two lines of information for Avstar NRCS and production personnel:

To see how the production cue example appears graphically, see “Title Entry Dialog Box” on page 9-3.

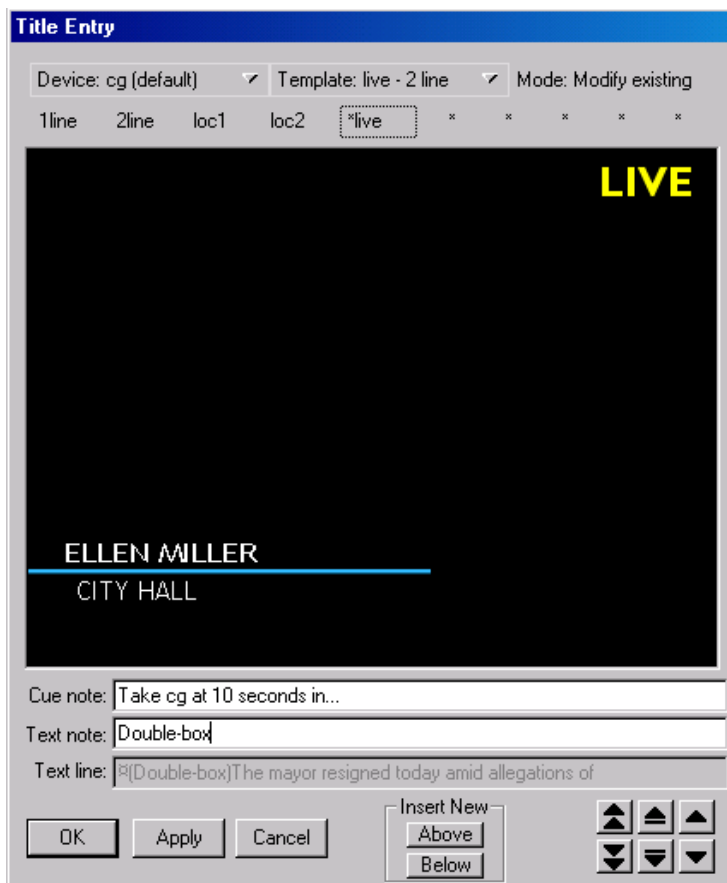
```
*CG live
Ellen Miller
City Hall
Take cg at 10 seconds in...
```

In the example, the user cannot visually determine how information will appear on air when the data is loaded in the character generator template. For instance, will the data fit into the actual text fields on the CG template? The answer to this question cannot be determined from the production cue text. However, the Title Entry feature allows for this possibility by offering a graphic-style dialog box.

The dialog box displays sample templates, complete with backgrounds and text fields, similar to actual CG templates that appear on air. These sample templates—which are configured and modified by system administrators using the CG Template Editor—can be filled in by the users, and information is applied to the production cue within the script. When configured accordingly, the Template Editor dialog box can provide users with visual verification of whether text they enter will fit in fields of actual character-generated graphics. Producers can use the Title Entry dialog box to preview CG information in scripts in order to check the appearance and accuracy prior to airing the production cue.

The following graphic shows how the previous production cue example appears in the Title Entry dialog box.

**Figure 9-1 Title Entry Dialog Box**



*The Title Entry feature can only be accessed by users when the cursor is located in the Story panel of the Avstar Workspace. Access to Title Entry can also be limited to certain users. See “Title Entry Security” on page 9-23 for more information.*

## Title Entry Setup and Configuration

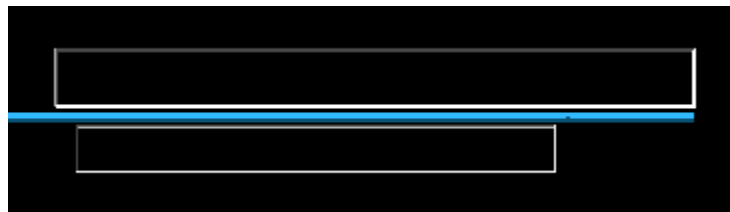
### Understanding CG Templates

Before you can use the Title Entry feature, templates must be created by the system administrator in Avstar to simulate the actual graphics used by the character generator. The person responsible for setting up the templates should have a good understanding of how CGs are created on the character generator, because it is important that templates in Title Entry correspond to those used by the character generator.

Most CGs are comprised of both background graphics and text fields. A background graphic could be a small color bar that spans the bottom third portion of the television screen, a full-screen graphic, a logo in the corner of the screen, and so forth. See “CG Template Backgrounds” on page 9-6 for more information.

The text fields vary as well. A CG template can contain numerous text fields, with each one having different predefined properties, such as screen location and font style. Typically, users can write text in the fields, but some may contain predefined text that users cannot edit. For instance, a CG can have a field with the predefined word, FILE, which would be aired over file video of previous news coverage.

An example of a standard CG template is a two-line lower-third CG. In the following example, the bar separating the two rectangular boxes make up the background. The boxes are the text fields, which can be filled in by a user.



CG text fields are also known as tab fields, referring to the key used to navigate from one field to another.

In a CG template, the character generator assigns each text field a number, indicating the order in which fields can be filled in. For instance, a user types a name in field one, then navigates to the second field in the sequence using the Tab key, and types the person's job title or location.



*Knowing the numeric field assignments of each template is crucial, because the sequence in the Title Entry dialog box must match the tab field's sequence on the corresponding character generator's template.*

CG templates are stored on the character generator at locations assigned or represented by identification numbers known as Template numbers. These numbers can be used by the CG operator to recall templates, which are filled in and aired during a show. However, in Avstar NRCS, aliases are usually assigned to each template to provide users with an easier way of recognizing the template.

For instance, a template containing one field—one line of text—used to air the location of a story may be stored on the character generator as template 502, but users type in the alias `loc1` in the production cue.

The assignment of aliases to Template numbers is stored in the Resource queue located in the System directory (`SYSTEM.RESOURCE`). It is recommended these same aliases be used in naming the templates for Title Entry.



**When a template is created, Avstar's CG Template Editor stores the configuration data associated with Title Entry in the Title-Entry queue in the System directory (`SYSTEM.TITLE-ENTRY`). This queue must be created, if it does not already exist, prior to creating templates. See "Creating a New Queue" on page 5-6 for more information. *Do not manually edit the data in this queue.* It is strongly recommended that, when configuring templates or setting up new ones, you use the CG Template Editor provided in the software. See "CG Template Editor" on page 9-11 for more information.**

## CG Template Backgrounds

Before creating CG templates for Title Entry, the system administrator should capture an image of the actual CG templates on the character generator as bitmaps. These bitmaps are used as background images in the creation process of corresponding templates in Avstar NRCS.

The Capture Tool is a program—included with your Avstar NRCS software CD—used to capture template backgrounds on character generators. It captures the templates as bitmaps with the exact specifications required for Avstar Title Entry templates.



*If you chose to use some other method of capturing bitmaps on the character generator, you must make certain to create bitmaps that are 400 pixels wide by 300 pixels high with 256 or more colors.*

You can store the bitmaps anywhere on the network as long as they are accessible from the Avstar Workstation used to run the CG Template Editor, the program used to create Avstar's Title Entry templates.

This section will explain what bitmaps are needed, hardware requirements for the Capture Tool, the installation process and procedures for using the Capture Tool.

## Required Bitmaps

Two slightly different bitmaps of each unique CG template should be captured from the character generator. First, capture the CG template with the text fields full of text.



*Do not use spaces. Each text field in the actual CG template filled out with information from a user should have a solid string of characters in it when the bitmap is made. Fixed-width fonts allow for absolute accuracy; however, proportional fonts can be used. The character limit of a field with fixed-width fonts will be the same no matter which letters of the alphabet are used to fill it up. Proportional—also known as variable-width—fonts contain some letters*

*that are wider than others, so the character limit may vary. For instance, the character limit of a field with a proportional font may be nine letters if “M” is used to fill up the field, but that limit increases to 25 when the letter “I” is used instead. A good mix of wide and narrow letters should be typed in the field to fill it up if a proportional font is used.*

The first bitmap provides a graphical demonstration of character width allowed in a text field and the position of that field on the screen. This bitmap will be used to align text fields and fonts on the Title Entry template when created in the CG Template Editor.

Secondly, capture the CG template with empty text fields as a bitmap. This bitmap will replace the first one in the final version of the Title Entry template. It will be used as the template background shown in the Title Entry dialog box.

## Capture Tool

The Capture Tool is a program used to capture graphical representations of templates on character generators. The tool captures a template as a bitmap, which can then be used as a background image in Avstar’s Title Entry templates.

### Hardware Requirements for Capture Tool

The hardware requirements to operate Capture Tool are:

- 133 Mhz IBM PC
- 32 MB RAM
- 10 MB Hard Disk space available
- XGA Monitor
- Osprey® 100 Video Card

## Installation of Capture Tool

For more information on the Osprey 100 video card, visit the Web site: [www.ViewCast.com](http://www.ViewCast.com).

Before you can install Capture Tool, the Osprey 100 video card should already be installed, according to the instructions provided with the card by ViewCast.com, Incorporated.

To install Capture Tool, do the following:

1. Run the setup program (`setup.exe`) from your Avstar NRCS CD. This program is on the CD in a subdirectory called GTECapture.
2. Follow any instructions as prompted in the program. The setup process will install the capture tool on your PC and create a Program Files group called Title Entry Capture Tool.



*You must provide a baseband video signal to the input of your Osprey card. This signal can be from a VCR or from your internal routing system. It does not require time-base corrected video; however, corrected video will produce better images.*

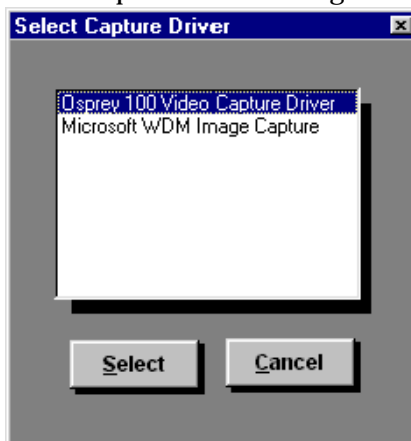
## Using the Capture Tool

To use the Capture Tool to capture a bitmap image, do the following:

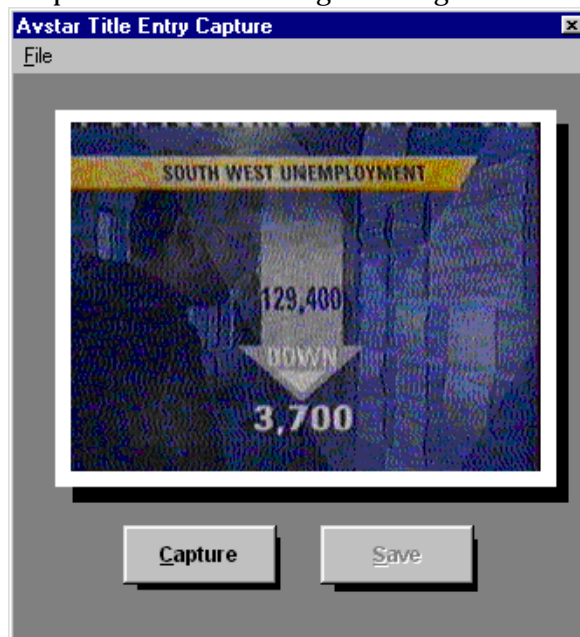
1. Click the Start button on the Task bar.
2. Select Program.
3. Select the Capture Tool program.



- When you start the Capture Tool, it will check to see if you have multiple video card drivers installed, and display a dialog box similar to the Select Capture Driver dialog box shown here:



- Select the driver for the card you will be capturing images from (typically, the Osprey card). Once you select the correct driver, the tool will open a window showing incoming video.

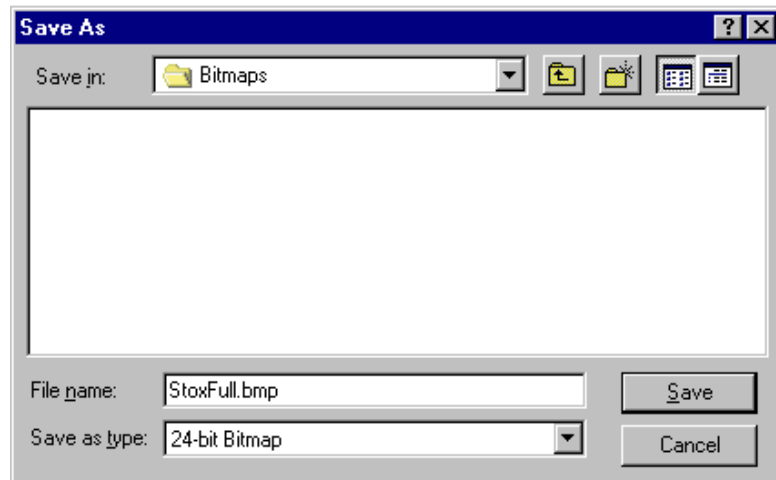


- Click the Capture button to freeze the image. Once the image is frozen, the Save button is activated.
- Click Save to save the image as a bitmap to be used in Avstar's CG Template Editor window.



*When using the Capture Tool, the image saved will be exactly 400x300 pixels, which is required for Avstar Title Entry templates. There is no need to adjust the size.*

- Select the location in the Save As dialog box where you want to save the bitmap.



Remember this location, which should be accessible from the PC running Avstar's CG Template Editor. You will refer to it later while using the CG Template Editor.

Repeat this procedure as often as needed to supply the necessary bitmap images for templates used by the Title Entry feature. Once the bitmaps are saved, you are ready to proceed in making Avstar CG Title Entry templates using the CG Template Editor.

## CG Template Editor

System administrators can use the CG Template Editor to create and modify templates referenced by the Title Entry dialog box. When a template is saved, its configuration information is stored in the System directory in the Title-Entry queue.



**iNews strongly recommends that you use the CG Template Editor to make all modifications. Do not manually edit the `SYSTEM.TITLE-ENTRY` queue.**

To start the CG Template Editor, do the following:

1. Sign on to Avstar NRCS.
2. Click on the Tools drop-down menu.
3. Select CG Templates. For superusers, the Edit Title Entry Template window will open.



*In some cases, a password is required before access to the CG Template Editor is allowed. See “Title Entry Security” on page 9-23 for more information.*

### Edit Title Entry Template Window

The Edit Title Entry Template window contains several template options, which can be used to create templates that graphically represent actual CGs aired from the character generator.

Some options available to the system administrator are:

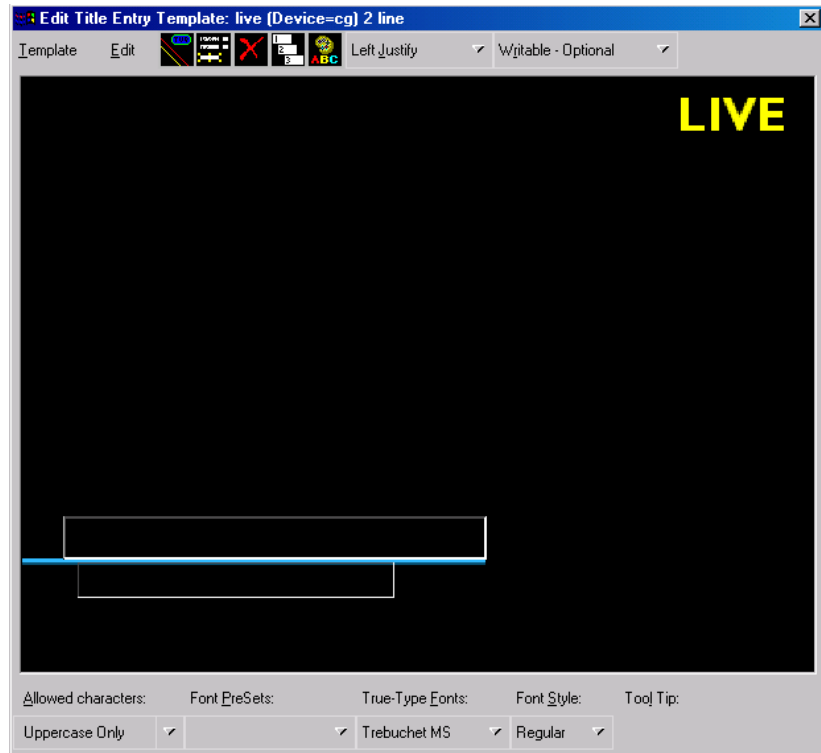
- Select a bitmap for a CG template background
- Add, modify, or delete text fields in a template
- Reorder sequence of text fields
- Limit editorial access to text fields
- Modify font size, color, style, and so forth

- Provide helpful Tool Tip text, which appears in the Title Entry dialog box when a user positions the mouse over a text field.








*Options in the CG Template Editor do not directly affect features of actual CG templates on the character generator.*

The Edit Title Entry Template window's toolbars—located at the top and bottom of the window—contain various buttons and drop-down menus, used to configure template options.



Here is an explanation of the toolbar buttons and drop-down menus:

**Table 9-1 CG Template Editor Toolbars**

Button or Menu	Options and Explanation
Template drop-down menu	Options include: New, Open, Save and so forth. This menu is used to open new or existing templates, save modified templates, and to close the window.
Edit drop-down menu	Options include: Background, Add Field, Font PreSets, and so forth. This menu is used for adding or deleting items to a template, such as a background or field, and setting predefined Fonts.
Logo button 	This button is the same as the Background option in the Edit drop-down menu. It allows the user to select a bitmap graphic as a template background.
Text Fields button 	This button is the same as the Add Field option in the Edit drop-down menu. It allows the user to add a text field to a Title Entry template, which can then be positioned and sized accordingly.
X button 	This button is the same as the Delete Field option in the Edit drop-down menu. It allows the user to delete the selected text field from a Title Entry template.
1-2-3 button 	This button is the same as the Field Order option in the Edit drop-down menu. It allows the user to rearrange the numerical tab sequence of the text fields in a Title Entry template.
Text Color button 	This button is the same as the Text Color option in the Edit drop-down menu. It allows the user to change the color of the font used in a text field as displayed in the Title Entry dialog box.

**Table 9-1 CG Template Editor Toolbars**

<b>Button or Menu</b>	<b>Options and Explanation</b>
Justify drop-down list	Options include: Left Justify, Right Justify, and Center Justify. This list is used to set the alignment of text in a field. The drop-down list will continue to display the current setting of the selected field when the list is closed. The default is Left-Justify.
Editing drop-down list	Options include: Writable, Read Only, Optional and Required. This list allows the user to determine whether a field is Writable or Read Only, and whether it is a required field. If Required is chosen, the user will be required to fill out the field with text before saving the production cue. Because the user is not prevented from entering text in a read-only field, if Read Only is chosen, you will be required to fill out the field with text before saving the template. Blank, read-only fields are not permitted. The drop-down list will continue to display the current setting of the selected field when the list is closed. The default is Writable-Optional.
Allowed Characters drop-down list	Options include: All characters OK, Uppercase Only, Lowercase Only, and Define Subset. This list allows the user to limit the type of characters allowed in a text field. For instance, if the text field should only contain numerical sports scores, then the user could use the Define Subset option to limit the allowed characters in that field to the numbers: 0123456789. As shown, do not use a comma to separate the numbers or it too will be allowed in the field. The Define Subset option is case-sensitive.

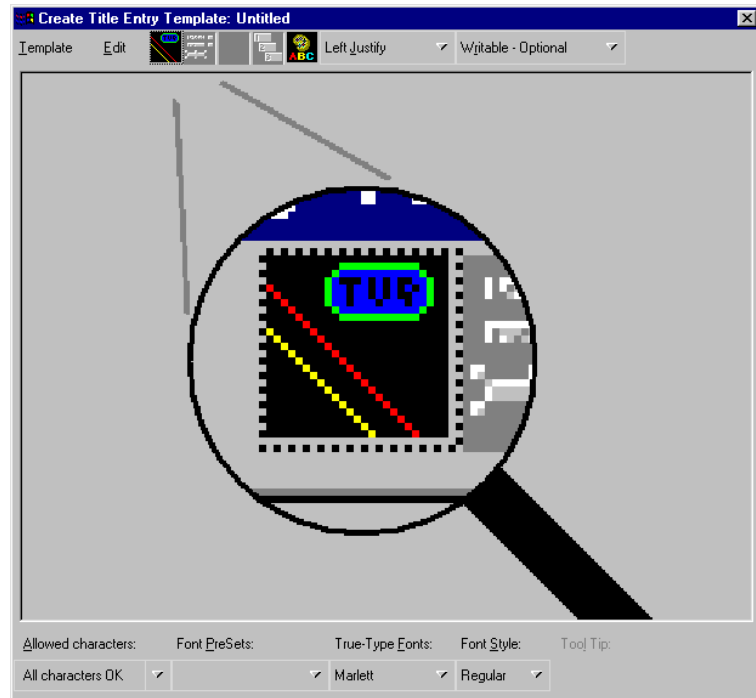
**Table 9-1 CG Template Editor Toolbars**

Button or Menu	Options and Explanation
Font PreSets drop-down list	Options include: 10 user-definable font display settings. This list allows the user to store up to 10 font display settings created by the user as Pre-Sets. These can later be selected when making new templates with fields that contain the same size and style of font, saving the user from having to recreate the font repeatedly. The first font stored is particularly important because it becomes the field properties default for the Template Editor at start up.
True-Type Fonts drop-down list	Options include: any true-type font installed on the workstation, such as Times New Roman. This list allows the user to select a font that matches or closely resembles the font used on the actual CGs.
Font Style drop-down menu	Options include: Bold, Italic, Underline, and Strike through. This allows the user to define the style of font used in the selected text field.
Tool Tip field	This field allows the user to type in text that will appear as a Tool Tip box in the Title Entry dialog box, when a user positions the mouse over the text field. For instance, if a user should type only a person's name or location in the field, the Tool Tip text for that field could be: Type Person or Location Name Here.

## Creating a New Template

Avstar's CG Template Editor is used to recreate a template that will simulate the appearance of information sent to actual templates aired by the character generator. This section assumes those actual CG templates were already captured as bitmaps and stored in a file accessible to the user creating a new template. See "Required Bitmaps" on page 9-6 for more information.

When the Create Title Entry Template window first opens, it appears with a graphic that highlights the first step in the process for creating new templates.



To create a new template from the Create/Edit Title Entry Template window, do the following:

1. Define a temporary background for the template by doing one of the following:
  - a. Click the Logo button.
  - OR-
  - b. Click the Edit drop-down menu, then select Background.
2. Select a bitmap from the directory where you stored them. This is the bitmap of the actual CG template you want to recreate in the

To create another new template when an existing one already appears in the window, use the Template drop-down menu, and select New.



Template Editor. See “Required Bitmaps” on page 9-6 for more information.



*The first bitmap you choose should be the one that contains text fields filled with characters. In upcoming steps, you will use this bitmap to align the text fields and the font in the template, before replacing it with another bitmap containing empty text fields.*

If a template already has a text field and it is selected, then any new field added to the template will follow size and font styles for that original field.

3. Add a text field by doing one of the following:
  - a. Click the Text Fields button.
  - OR-
  - b. Click the Edit drop-down menu, then select Add Field.
4. Click on the new text field and drag it into position over the text-filled area pictured in the bitmap—one field per line of text in the bitmap. You can also adjust the position of the text field by using any of the following key combinations:
  - **CTRL-Right Arrow** - - - -Move field to the right
  - **CTRL-Left Arrow** - - - -Move field to the left
  - **CTRL-Up Arrow** - - - - -Move field up
  - **CTRL-Down Arrow** - - - -Move field down
5. Use the mouse to resize the text field to reflect the size of the text area in the bitmap.
6. Use the True-Type Font drop-down list or the PreSet list to select a font similar to the font used by the character generator, as seen in the bitmap.
7. Type the same characters shown in the bitmap into the new text field. The font may or may not line up exactly over that of the bitmap. To enable Title Entry to accurately reflect the text as it will appear on actual CGs, you must align the field and characters with those shown in the bitmap.
8. Adjust the height and width of the font by using the following key combinations:

- **ALT-Right Arrow** - - - - Expand width of font
- **ALT-Left Arrow** - - - - - Contract width of font
- **ALT-Up Arrow** - - - - - Expand height of font
- **ALT-Down Arrow** - - - - Contract height of font

This may take some time and will require experimentation with fonts and font styles, before the characters align with those of the bitmap. The goal is to make certain the text field you create in the template has the same character limit as the area shown in the bitmap when the exact characters are used in both.

9. Use the backspace key to erase characters from the field once the font and field alignment is complete.
10. Repeat steps 5-11 as needed for each text field in the template.



*If text fields on several CG templates use the same font and field size, you do not have to repeat steps 9 and 10 with each new template created. You can duplicate a field (including all its size and font properties) within a template by holding the Control (CTRL) key down, clicking on the field, and dragging its copy to a new location within the template. You can also reuse field properties by saving them as a Font PreSet, which can then be selected for use when creating other templates. See “Using Font PreSets” on page 9-20 for more information.*

11. Determine the justification of text in each field. Select the field and use the Justify drop-down list.
12. Determine whether a user can write in each field, and whether the user is required to fill in a field by using the Editing drop-down list. If you make a field required, then the user must enter some data into that field when using Title Entry before Avstar NRCS will accept the production cue. This does not apply to the field while in the CG Template Editor. However, if you make a field read-only, it must contain some text so you will be required to put some text in the field while in the CG Template Editor. Users will be prevented from altering that text in the read-only field when using Title Entry.

The Read Only option can be used for setting up templates that standardize certain commonly used CGs. For instance, you have a CG that has two fields: one for any council member's name and the second that should contain the words, City Council Member. But, users sometimes just write Councilman or worse, they misspell Council. You could make the second field read only and pre-define the words that appear in the field.



*If a field is pre-defined on the actual CG template, such as LIVE or FILE, it should appear as part of the background—saved as part of the bitmap—not as a read-only field on the Title Entry CG template.*

The Allowed Character option is case-sensitive. Also, notice the example in step 15 does not include commas, such as: 1, 2, 3, 4, OT, F. If commas are included, then commas will also be accepted in the field.

13. Determine which characters are allowed in each text field. Select the field, then choose from the options in the Allowed Characters drop-down list. If you choose to define your own subset of characters, a dialog box will appear, where you must then type in only those characters you want to allow in the field. For instance, you can limit what a user types into a field for a Sports CG to only characters: 1234OTF—representing 1st-4th quarter, OT for over-time, and F for final.
14. Type in the text for the field's Tool Tip. The Tool Tip will appear in the Title Entry dialog box when the user positions the mouse over that field. To enter Tool Tip text, click on the Tool Tip field at the bottom-right corner of the Edit Title Entry Template window, then type your text in the field provided. You must press Enter to save the tool tip text.



*ALT-L can be used to select and enter data into the Tool-Tip field. Also, each Tool Tip will appear with a number in the CG Template Editor window. The number indicates that field's order in the template. These numbers do not appear as part of actual Tool Tips over fields of the Title Entry dialog box.*

15. Replace the background of the template with the bitmap of the CG Template containing empty text fields. To chose another bitmap, do one the following:
  - a. Click the Logo button.  
-OR-
  - b. Click the Edit drop-down menu, then select Background.
16. Save the template by doing one of the following:
  - a. Click the Template drop-down menu, then select the Save option.  
-OR-
  - b. Type CTRL-S.



**The template name must match the alias associated with the actual CG template number used by the character generator. Aliases are defined in `SYSTEM.RESOURCE`. For instance, if a user typically types `*CG loc1` as the first line of a production cue for a 1-line location CG, then the Title Entry template created for that CG should also be named `loc1`. The alias must be defined before opening the CG Template Editor.**

## Using Font PreSets

Font PreSets are a time-saving feature for system administrators using the CG Template Editor to create Title Entry templates.

Once a system administrator sets up a text field and saves it as part of a template, he can store settings for that field as a Font PreSet. The Font PreSet can then be selected when the system administrator needs to add a field with those same settings to another template. For instance, the text field of a one-line location CG is saved as a Font PreSet, which is later used to create the first field in a two-line name CG, a two-line live CG, and so forth, because all fields use the same font style and field size.

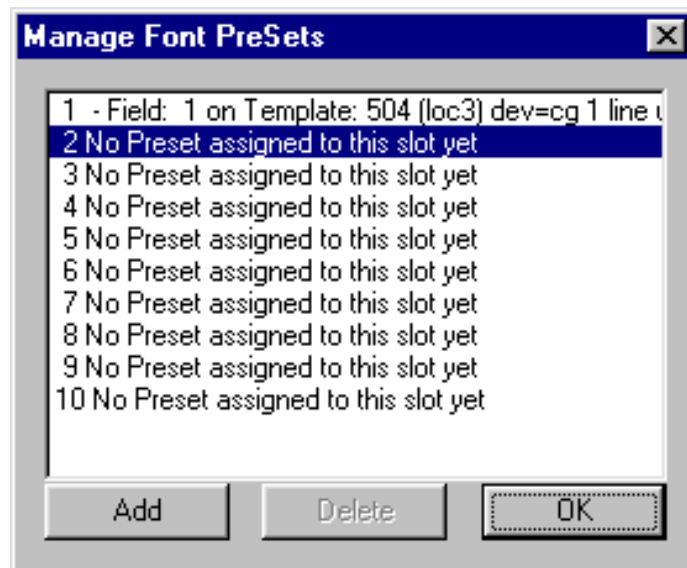


*You can have a maximum 10 Font PreSets in Avstar NRCS. The first one stored is particularly important, because it becomes the field properties default for the CG Template Editor at start up. Save the PreSets for major font/field variations. Because you are limited to 10, it is not recommended they be used for minor variations, such as a font color change. You can replace any Font PreSet with a new one if all of the PreSets are taken.*

To save a field as a Font PreSet, do the following:

1. Open the template containing the field you want to save as a Font PreSet.
  2. Click on the specific field to select it.
  3. Do one of the following:
    - a. Click the Edit drop-down menu and select Font PreSets.
- OR-
- b. Press the F7 key.

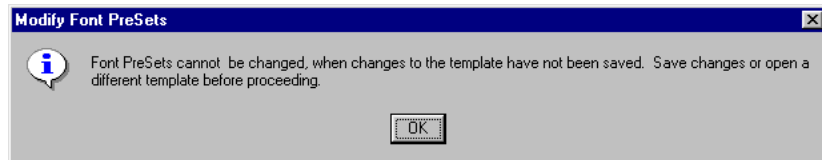
The Manage Font PreSets dialog box will appear, with the selected field placed in the first available PreSet option.



4. Click Add to save the selected field as a new Font PreSet.
5. Click OK to close the dialog box.



*You cannot save a field's settings as a Font PreSet until after the field is saved as part of a template. If you attempt to set a Font PreSet using a newly created field from an unsaved template, Avstar NRCS will issue the following advisory.*



To delete an existing Font PreSet, do the following:

1. Open the Manage Font PreSets dialog box, by doing one of the following:
  - a. Click the Edit drop-down menu and select Font PreSets.

-OR-

  - b. Press the F7 key.
2. Select the PreSet you want to delete from the list.
3. Click the Delete button.
4. Click OK to close the window.

To choose an existing Font PreSet for a selected text field, do the following:

1. Click the Font PreSets drop-down list at the bottom of the Edit Title Entry Template window.
2. Select the Font PreSet you want to use for the chosen field.
3. Press Enter.

## Title Entry Security

Security to CG Template Editor and CG Title Entry is determined separately per user account. A user who can use CG Title Entry will not necessarily have access to CG Template Editor, while another user can be denied the ability to use both features.

### Access to CG Template Editor

Access to CG Template Editor is limited to system administrators, or superusers, and those who know the password for `dbmanager`, if that user account is created by a system administrator. This is the same `dbmanager` account that is used to modify database traits. See “Creating a Database Manager Account” on page 4-36 and “Changing Database Traits” on page 5-21 for more information.

When a non-superuser attempts to launch the CG Template Editor from the Tools drop-down menu, the following dialog box will appear:



If the correct password is not given, then Avstar will notify the user and deny access to the Edit Title Entry Template window.



*Only one person on the network can use CG Template Editor to edit Title Entry templates at a time.*

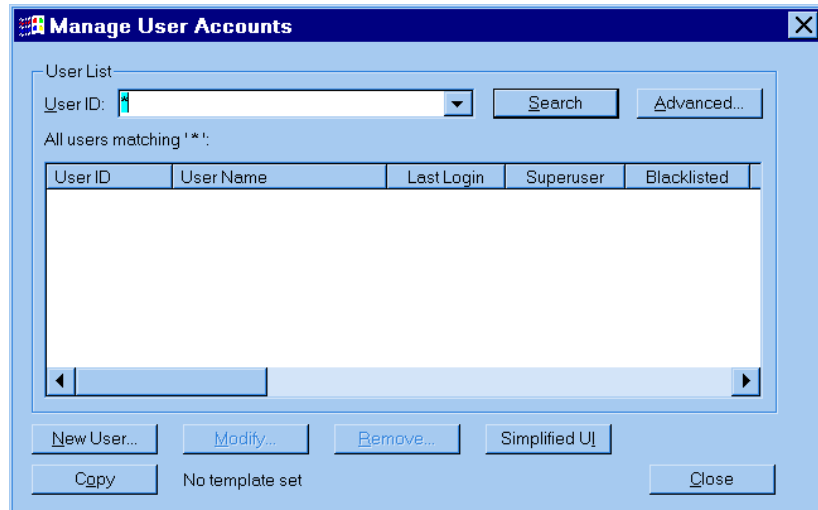
### Access to CG Title Entry

The availability of CG Title Entry is dependent on two things: cursor position in the Avstar Workspace and permission granted in the user’s account.

When a user's cursor is in any panel other than the Story panel, the Titling option in the Tools drop-down menu will appear gray, indicating that Title Entry is unavailable. Additionally, a system administrator, or those authorized with the `umanager` password, can deny a user's access to Title Entry.

To prevent access to Title Entry for a specific user, do the following:

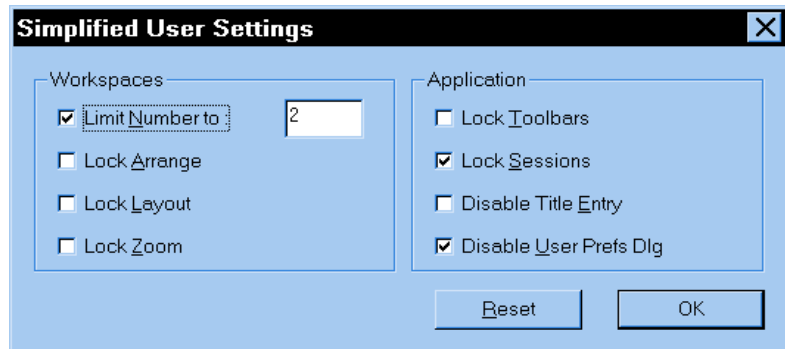
1. Sign on as a system administrator, or with your own user account if you know the `umanager` password.
2. Click the Tools drop-down menu.
3. Select Options.
4. Select Users. System administrators will see the Manage User Accounts window open. Others will be prompted to give the `umanager` password first before the Manage User Accounts window will open.



5. Indicate the user, by providing the User ID. The Search button is available to help locate the user's account if the ID is unknown.



- Click the Simplified UI button. The Simplified User Interface dialog box will appear.



- Click on the Disable Title Entry checkbox to select it.
- Click OK.



# CHAPTER 10

## *ed, the Line Editor*

As an Avstar system administrator you will need to know how to use the UNIX line editor, called `ed`, to make changes to important system files. These include the system configuration, system profile, and printer profile files.

This chapter contains:

- Overview - Before Editing
- Using the UNIX Line Editor
- Launching `ed`
- Editing Commands
- Saving Changes
- Quitting `ed`

## Overview - Before Editing

Configuration of the Avstar Newsroom Computer System (NRCS) is controlled by settings stored as text files in the (root) Site directory on the software area of the system's hard disk. Changes to system configuration are made by editing these text files using the UNIX line editor, `ed`.

However, before editing a system file, you can view, print, and as recommended, make a backup copy of the file.

### Making a Backup File Before Editing

When you want to make changes to a system file, begin by making a backup copy of the file, and then edit the backup file. That way, if you make a mistake during the editing process, your original file version is preserved.

To copy a file, use this format:

```
cp <file pathname> <new pathname>
```

For instance, to copy the configuration file in the Site directory, type:

```
AVSTAR-A: cp /site/config /site/config.test
```

### Viewing the Contents of a File

To view a system file, use the `cat` command at the console. The format is:

```
cat <file pathname>
```

For instance, to view your copy of the configuration file, created in the previous section's example, type:

```
AVSTAR-A: cat /site/config.test
```

Information similar to the following appears:

```
host ab a
net 125 10
servers 171
;
host ab b
net 125 20
;
host a a
net 125 10 20
servers 171
;
host b b
net 125 10 20
servers 171
;
pcu 10 pcu10 at 11 12 13 14 15 16 17 -
workstation 11 19200-72 news - ;[1-01] 6p
workstation 12 9600-71 news - ;[2-01]
workstation 13 9600-71 news - ;[2-06] 7p-ad
workstation 14 9600-73 news - ;[3-03] 10p-wtr1
line 15 2400-7e modem hayes ;[1-07] modem
line 16 1200-7e modem hayes ;[1-02] modem
line 17 1200-7e modem hayes ;[4-04] TI880
;
pcu 20 pcu20 at 21 22 23 24 25 26 27 28
workstation 21 9600-7e 2 news - ;[1-02] 6p
...
```

## Printing a Copy of a File

To send a file to a serial printer instead of viewing it on the console screen, precede the `cat` command with `print` and the number of the printer you want to use. Use this number to refer to that printer when you print from a computer connected to the Avstar system.

For instance, to print `/site/config.test` on printer number 1, type:

```
print 1 cat /site/config.test
```

## Using the UNIX Line Editor

The UNIX editor, also called `ed`, is a line editor. Unlike Microsoft® Word or similar word processing programs, `ed` deals with text files on a line-by-line basis.

To learn more about the UNIX line editor, refer to one of the books available in bookstores on the UNIX operating system with sections devoted to `ed`.

The most likely reason for you to use `ed` is to modify system files, such as `/site/config` or `/site/etc/rc`. For instance, if you add a workstation to your system, you will need to add the computer's configuration information to the configuration file in the Site directory. Because many system files are located in the Site directory, they are also referred to as site files.

Other system files (in Site directory) or subdirectories include:

- `/site/system`
- `/site/wires/<file name>`
- `/site/printers/<file name>`
- `/site/dict`
- `/site/mcs`



*When you modify a site file, make the same changes to each computer's copy of the file, or your system will not run properly. Select all computers before you open a file for editing to ensure changes you make are applied to each computer's copy of the file.*

## Launching `ed`

To launch `ed` from the Avstar console, do the following:

Type `ed` followed by the file path and name to be edited. For instance, to edit a copy of the configuration file in the Site directory, you would type:

```
AVSTAR-A: ed /site/config.test
3624
```

The UNIX editor, `ed`, returns a number indicating the file size expressed as the number of characters, including spaces and returns. Also, the console prompt, `AVSTAR-A:`, changes to no prompt when you launch the editor.

If the file name specified does not exist or is a non-text file unsuitable for editing with `ed`, `ed` returns a question mark (?) followed by the file name. This is one way to create a new text file. For instance, a new text file called `newfile` is created when the following is typed:

```
AVSTAR-A: ed newfile
?newfile
```



**Do not attempt to edit a non-text file such as a binary file. Doing so could cause undesirable results.**

## Specifying Lines to Edit

The line you are on presently is called the current line.

Because `ed` is a line editor, you navigate through the file by line numbers. For instance, when you open a file for editing, `ed` considers the

last line in the file the current line. If you want to view or edit a different line, you must go to that line.

For instance, you can move to the first line of the file by typing **1** and pressing Enter. To move ahead five lines, you could type **+5**. To move back three lines, you could type **-3**.



*The UNIX line editor, ed, will respond with a question mark (?) if you try to move beyond the last line of the file. Additionally, you cannot type a minus (-) value greater than or equal to the current line number, because you cannot move to a line preceding the first line in the file.*

Within ed, pressing Enter with no line number reference or command will cause ed to make the next line in the file the current line, displaying that line as it goes. For instance, in the following example, the user selects line 19 in the file, then presses the Enter key three times. The UNIX line editor, ed, responds each time by displaying lines 20, 21, and 22, respectively.

```

19
;printer    11    2400        11          -    ;
Enter
terminal    12    19200        0          news    -    ;
Enter
terminal    13    9600-8n      1          news    -    ;
Enter
wire        14    2400          anpa7     AX     -    ;

```

When editing, it is necessary to specify the line number(s) to be acted upon. This can be done in several ways:

- Type the line number.
- Type starting and ending line numbers separated by a comma.
- Type period (.) to specify current line number.
- Type a dollar sign (\$) to specify last line in the file.



The editing command to act upon the specified line(s) is typed immediately following the specified line(s). There should be no spaces. Here are some examples using the Print command, `p`.

To make line 18 the current line, type:	<code>18</code>
To print (to screen) line 10 of the file, type:	<code>10p</code>
To print (to screen) lines 10-20, type:	<code>10,20p</code>
To print all lines from the current line to line 20, type:	<code>.,20p</code>
To print all lines in the file from line 80 to the end, type:	<code>80,\$p</code>
To display the current line number, type:	<code>. =</code>
To display the line number of the last line, type:	<code>\$ =</code>
To make the fifth line from the bottom current, type:	<code>\$-5</code>

## Searching the File

When you do not know the line number, but you want to locate a line containing a specific word, phrase, or number, you can use the search option. The UNIX line editor, `ed`, will search the file, starting at the current line, and display the line with the next occurrence of the specified text.

To search for text, do the following:

1. Type a forward slash (`/`). Do not press the spacebar.
2. Type the text you want to locate, followed by another forward slash.
3. Press Enter.

For instance, if you want to find `websession` in the configuration file, type:

```
/websession/
;websession      900
```

In the previous example, `ed` found the word, `websession`, on line 900. If you want to repeat the search to locate further occurrences of that word, type a forward slash and press Enter again. For instance:

```
/websession/  
;websession      900  
/  
;websession      901  
/  
;websession      903
```

In the previous example, the user repeated the original search command two more times. Each time, `ed` responded with the word searched and the line number where the next occurrence of the word appears. In each case, the current line becomes the line number displayed by `ed`.

### Searching Tips

Here are a few more tips for searching with `ed`.

- Remember to use spaces before and/or after text to further define your search string. For instance, type `/ 25 /` instead of `/25/` to avoid finding other numbers that contain the number 25, such as in the line: `net n125 20`
- Remember that searches are case sensitive. For instance, searching for `/PCU/` does not find `pcu`.
- Remember that searches distinguish between spaces and tabs. In other words, if you use spaces and the file contains the information separated by a tab, you will not find it. For instance, the search example below will not work:

```
You type: /pcu 20/      (pcu and 20 are separated by a space)  
The line is: pcu      20  (pcu and 20 are separated by a tab)
```

## Editing Commands

There are several basic editing commands you can use in `ed` to view, change, add, move, and delete text in a system file.



*Some commands, such as Add, Delete, and Insert, change the current line, while others do not. For instance, the Print command sets the current line to the number of the last line printed.*

Here is a list of editing commands, along with examples of their use:

Command	Description & Examples
<code>a</code>	<p>The Append command inserts one or more lines <i>after</i> the selected line. For instance:</p> <pre> /websession/ ;websession      900 a anws              511      -   1      gnews      -   ; anws . </pre> <p>In the above example, the user searches for <code>websession</code>, and <code>ed</code> responds by displaying line 900, the first line found containing that word. The user types <code>a</code>, presses Enter to start the append operation, and types information to be inserted in the file <i>after</i> line 900. The user then types a period (<code>.</code>) on a line by itself, which is very important because it terminates the append operation. Without it, successive lines typed by the user would be added to the file.</p>

Command	Description & Examples (Continued)
c	<p>The Change command replaces the entire contents of the line addressed. For instance:</p> <pre>21 terminal 13 9600-8n 1 news - ; c ; not used .</pre> <p>In the above example, the user selects line 21, and <code>ed</code> responds by displaying line 21. The user types <code>c</code> on one line to start the change operation. On the second line, the user types replacement text and presses Enter, followed by a period on a line by itself. The period (<code>.</code>) is very important because it terminates the change operation. Without it, successive lines typed by the user would be added to the file.</p>
d	<p>The Delete command is used to delete the line(s) specified. For instance:</p> <pre>27d          - Deletes line 27. 30,35d       - Deletes lines 30 through 35. 40,\$d       - Deletes every line from 40 to the end of the file.</pre>
g	<p>The Global command allows the user to apply an editing command to all lines in the file that contain a specific word, phrase, or number. For instance:</p> <pre>g/anws/s/anws/asws - Finds all occurrences of anws and                     uses the Substitute command to                     replace anws with asws.  g/websession/d     - Finds all occurrences of                     websession and deletes the lines                     containing it.</pre> <p>Each line affected is displayed after the editing command is applied.</p>

Command	Description & Examples (Continued)
c	<p>The Change command replaces the entire contents of the line addressed. For instance:</p> <pre>21 terminal 13 9600-8n 1 news - ; c ; not used .</pre> <p>In the above example, the user selects line 21, and <code>ed</code> responds by displaying line 21. The user types <code>c</code> on one line to start the change operation. On the second line, the user types replacement text and presses Enter, followed by a period on a line by itself. The period (<code>.</code>) is very important because it terminates the change operation. Without it, successive lines typed by the user would be added to the file.</p>
d	<p>The Delete command is used to delete the line(s) specified. For instance:</p> <pre>27d          - Deletes line 27. 30,35d      - Deletes lines 30 through 35. 40,\$d      - Deletes every line from 40 to the end of the file.</pre>
g	<p>The Global command allows the user to apply an editing command to all lines in the file that contain a specific word, phrase, or number. For instance:</p> <pre>g/anws/s/anws/asws - Finds all occurrences of anws and                     uses the Substitute command to                     replace anws with asws.  g/websession/d     - Finds all occurrences of                     websession and deletes the lines                     containing it.</pre> <p>Each line affected is displayed after the editing command is applied.</p>

Command	Description & Examples (Continued)
i	<p>The Insert command inserts one or more lines <i>before</i> the selected line. For instance:</p> <pre data-bbox="515 392 1179 574"> /websession/ ;websession      900 i <b>anws</b>           511      -    1    <b>gnews</b>    -    ; <b>anws</b> . </pre> <p>In the above example, the user searches for <code>websession</code>, and <code>ed</code> responds by displaying line 900, the first line found containing that word. The user types <code>i</code> and presses Enter to start the insert operation. The user then types two lines of information to be inserted in the file <i>before</i> line 900. On the last line, the user types a period (<code>.</code>) on a line by itself, which is very important because it terminates the insert operation. Without it, successive lines typed by the user would be added to the file.</p>
m	<p>The Move command removes the line(s) specified from their original location and inserts the line(s) <i>after</i> the target location. For instance:</p> <pre data-bbox="515 968 1222 1069"> <b>18m20</b>          - Line 18 becomes line 20. Lines 19 and 20 become                     18 and 19. <b>1,5m\$</b>         - Moves lines 1 through 5 to the end of the file. </pre>
p	<p>Contrary to the name, the Print command does not send information to a printer. Instead, it <i>prints</i> text to the console screen. It is handy for viewing specific lines within a file. Typed alone with no line number references, the Print command displays the current line.</p> <p>For instance, to print (to screen) lines 10-20, type: <b>10,20p</b>.</p> <p>More examples of the Print command can be found in “Specifying Lines to Edit” on page 10-5.</p>

Command	Description & Examples (Continued)
s/<old text>/new text>	<p>The Substitute command substitutes a specific portion of a line with new text.</p> <p>For instance:</p> <pre> 3 ;AvidNews Starter config 09JAN00 <b>s/AvidNews/Avstar</b> ;Avstar Starter config 09JAN00 </pre> <p>In the above example, the user selects line 3, and ed responds by displaying line 3. The user substitutes the word <code>AvidNews</code> for <code>Avstar</code> but does not alter anything else on the line. The UNIX line editor, ed, confirms the substitution by redisplaying line 3, incorporating the change.</p>
t	<p>The Copy command copies the line specified, and inserts a copy <i>after</i> the target location. For instance:</p> <pre> 5t10      - Inserts a copy of line 5 <i>below</i> line 10. The copy            becomes line 11. The original line 11, if any,            becomes line 12, and so forth. 20,30t50  - Inserts copies of lines 20 through 30 <i>after</i> line 50. </pre>
u	<p>The Undo command is used to cancel the effects of the last editing command entered. For instance:</p> <pre> 1,5m\$ u </pre> <p>In the above example, the user issues a command to move lines 1 through 5 to the end of the file, then types <code>u</code> to undo that command.</p>

## Saving Changes

The changes you make to a file are not saved immediately. This means you could quit (or exit) `ed` without saving changes if necessary. You must use the Write command to save modifications.

To save changes, type `w` and press Enter. The UNIX line editor, `ed`, will respond by displaying the file size, such as:

```
w
3624
```



**The Write command is case-sensitive. If uppercase `W` is used, `ed` will *append* the modified version of the file to the end of the original file version. This can quickly increase the file size and result in redundancy. Always use the lowercase `w`.**

## Quitting `ed`

To exit the UNIX editor, type `q` and press Enter. For instance:

```
w
3624
q
AVSTAR-A:
```

In the previous example, the user saved changes first by using the Write command, then typed `q` to quit `ed` and return to a normal console prompt.

However, if you wanted to quit `ed` without saving your changes, you can do so. In this case, you would have to type the Quit command twice: the first time to notify `ed` you want to quit, and the second time to confirm that you want to quit without saving changes.



For instance:

```
q
?
q
AVSTAR-A:
```

In the previous example, `ed` responds to the first `Quit` command with a question mark (?) to remind the user changes were made to the file and not saved. This is a precautionary warning to help prevent a user from exiting `ed` and inadvertently losing changes that were not saved. When the user replies by typing the `Quit` command a second time, `ed` exits, abandoning any changes made.



# CHAPTER 11

## *Configuration Files*

Information about your Avstar system's hardware devices (workstations and printers), connections (wires and links), and Avstar Servers is stored in *configuration files*. A representative sample of each Avstar configuration file is contained in this chapter. A comprehensive list of sample configuration files is contained in Appendix B. Procedures for editing configuration files are covered in this chapter using the UNIX line editor, `ed`. Instructions for using `ed` are found in Chapter 10.

This chapter has the following sections:

- Licensing of Avstar System Components
- Device Types
- Viewing Information About Your Devices
- The Site Configuration File (`/site/config`)
- The Hosts File (`/etc/hosts`)
- System Profile Files
- Adding Devices to Your Avstar System
- Editing the Site Configuration File in the Database
- Intersystem Messaging

## Licensing of Avstar System Components

Each time the Avstar Newsroom Computer System is configured, your licensing information is checked. This information determines the number of devices you are authorized to connect to the system. An error message appears if the configuration file defines more devices than are licensed in any of the following categories:

- All serial devices (printers, wires, and so forth)
- PCUs and CCUs
- Avstar and network workstations

To display your system's licensing limits, at the Avstar console, type:

AVSTAR-A: **status license -OR- status 1** (lowercase L)

A message similar to the following will appear on your screen:

```
A is ONLINE and has been CONFIGURED. ID is AVSTAR
System is AB. Master is A.
Disk status is OK
Site Key: 001234
CPUs: 2
PCUs: 10
Serial devices: 100
Network Terminals: 30
DEC Servers: 2
LAT Printers: 2
Avstar News DOS machines: 100
Avstar News DOS resources: 100
Avstar News Windows machines: 3000
Avstar News Windows resources: 1000
Avstar News Web Access sessions: 40
Sony Barcode Printer allowed.
```

The *machines* category defines the number of workstations authorized to connect to Avstar Servers and receive the login prompt. These authorized workstations are entered in stories located in `SYSTEM.CLIENT.WINDOWS` and `SYSTEM.CLIENT.DOS`. The *resources* category defines the total number of simultaneous login sessions.

To change license allowances, contact an Avstar Systems sales representative.

## Device Types

Avstar supports the following types of devices:

- *Workstation* – Any device on which a user can log in to the Avstar Newsroom Computer System. This includes Avstar Workstation PCs, DOS PCs, terminals, and dialup modems.
- *Printer* – A device that prints stories and scripts. Avstar NRCS supports serial and Windows printing. Avstar NRCS also supports the ability to print to a queue in the Avstar database.
- *Wire* – A device that processes wire service stories in the database.
- *Service* – A device that allows a user to connect to another computer system from a workstation.
- *Server* – A utility program that performs tasks on stories in a queue, based on defined instructions. Supported server types are action, distribution, parallel wire, and keyword. Other servers facilitate searches, mail, and printing. See “Servers” on page 14-1 for more information.
- *Rx/Tx Network Link* – Device designed for receiving or transferring stories between computer systems.

The server listed here is not the same as the computers used as Avstar Servers, typically given names with the station’s call letters and an A, B, or C. For instance, AVSTAR-A.

## Viewing Information About Your Devices

To view information about devices connected to your system, use forms of the `list c` console command. The console command `list c` prints information to the console screen about configuration of a device or devices.

The format for this command is:

```
list c [<device type, device number, or program name>]
```

The device type or device number is optional. If you do not enter them, you get a list of the configuration information for all devices in your system.

When you type `list c` at the console, information similar to the following appears:

DEV	DEVICE_TYPE	COMPUTER	CCU	PRINTER	SPEED	OPTIONS	DEVNAME
C60	ccu	A-AVSTAR	60			P	
T61	news	A-AVSTAR	60-1	3	9600	8N	
S200	monitor	A		N200			
S401	txnet	A-net		N401			txnet
...							
S522	seek	A		N522			

If you follow `list c` with the name of a program, Avstar lists every device on your system that uses that program.

For instance, to find out how many devices use the action server program, type:

```
list c action
```

Information similar to the following appears:

DEV	DEVICE_TYPE	COMPUTER	CCU	PRINTER	SPEED	OPTIONS	DEVNAME
S344	action	A		N344			
S345	action	A		N345			

To list configuration information for device number 14, type:

```
list c 14
```

A message similar to the following appears:

DEV	DEVICE_TYPE	COMPUTER	CCU	PRINTER	SPEED	OPTIONS	DEVNAME
T14	news	A-ccu10	10-4	1	19200		14

## List C Message Columns

There are eight columns in the `list c` messages.

- DEV
- DEVICE\_TYPE
- COMPUTER
- CCU
- PRINTER
- SPEED
- OPTIONS
- DEVNAME

Each column is explained in this section.

**DEV** – Lists the device number. The number is preceded by a capital letter identifying the type of device.

The device identifiers (the first element of the display under **DEV**) are defined as follows:

Identifier	Supported Device
B	Web browsers
C	CCU
D	Dialup line

Identifier	Supported Device
G	Avstar Workstation graphical user interface
L	Serial resource (line)
M	MCS driver, MCS PC
N	Avstar for DOS PC
P	Printer
R	Network resource
S	Server, special
T	Terminal, network terminal
U	Unused
W	Wire

**DEVICE\_TYPE** – The program that runs the device. For instance, workstations use the *news* program and have *news* listed in this column.

**COMPUTER** – Identifies the computer to which a CCU/PCU (or a device, through its CCU/PCU) is attached. For instance, a network PCU running on computer A and using the network name `pcu10` would have `a-pcu10` in this column.

**CCU** – Contains the CCU/PCU device number and port on that CCU/PCU to which the device is attached. For instance, a device connected to port 4 on PCU 20 has `20-4` in this column. If the device is a CCU/PCU, only its device number appears here.



- PRINTER** – Lists the printer number assigned to the device. For instance, a terminal assigned printer number 2 would have 2 in this column. Wires display a `join` or `raw` option.
- Servers and specials use this column to indicate which mailbox they use to receive notifications. For instance, a `txnet` link with notification mailbox 189 would have `N189` in this column.
- SPEED** – Indicates the bps rate used for communication between a device and its CCU/PCU.
- OPTIONS** – Lists any modifiers to the device's speed, such as bits per character, parity, and handshaking. For CCUs/PCUs, an A for CCU or a P for PCU is listed.
- DEVNAME** – Lists the device's name, if it has one, which can be used for group membership and placed in the `DEVNAME` field of stories.

## The Site Configuration File (`/site/config`)

The site configuration file (`/site/config`) is a system road map. It lists all devices, servers, and resources configured to run on your system and how they are connected. If a device is not in the site configuration file, the system will not know about it and cannot use it. Standard devices and resources you may configure in this file include terminals, printers, Avstar Workstations, and wire services.

Each Avstar Server for your system has a copy of this file that it reads when it starts up and when you execute the `configure` console command. However, it is only the configuration file on the master computer that is active and used when the system is started up.



Whenever you make changes to a site file, such as the configuration file, be sure to select *all* servers in your system at the Avstar console. Unlike database stories, site files are not automatically mirrored from one computer's disk to another. See "Selecting Servers" on page 2-5 for more information.

Table 11-1 shows some of the more common device configuration lines.

**Table 11-1 Sample Device Configuration Lines**

Type	Number	Speed	Printer	Program	Name	Comment
anws	301	–	1	gnews	–	;Avstar PC Workstation
andos	411	10.2.1.34	1	news	–	;Avstar for DOS Workstation
terminal	23	19200-7e	5	news	–	;palmer
terminal	100	0	–	news	–	;network terminal
dialup	47	38400-8na	0	news	–	;555-1212
printer	37	9600	4		–	;laserjet4
wire	38	9600	anpa7	AP	–	;AP wire
line	42	38400-8na	modem	hayes		;dialout
line	42	38400-8na	radio	direct		;serial to WXYZ radio
resource	226	console	–			;net connect
server	211	monitor	211	–		;6pm show
server	233	action	233	–		;action server

**Table 11-1 Sample Device Configuration Lines (Continued)**

Type	Number	Speed	Printer	Program	Name	Comment
server	234	mail	234	-		;mail server
server	235	action	-	-		;timed action
special	51	19200	mct	-	200	;MCTerminal
special	220	0	txnet	220	-	;txnet to archive
special	55	9600	telex	55	telxl	;telex
driver	45	9600-8nh	infindriver	CG1		;Chyron 1
driver	62	9600-8nh	qpboxdriver	SS		;Picturebox
mcspc	80	mcspc1	infindriver	CG2		;Chyron 2

## Testing the Site Configuration File After Changing

Whenever you make changes to `/site/config`, always run a test on the changes to ensure there are no problems with the new configuration. By doing so, the test will warn of problems or if license limits are exceeded. Some configuration problems will prevent system configuration and startup.

To run the test, use the `configure` console command in the following format:

```
configure <config file> <system> <computer>
```

The following example tests configuration for the A server in an AB system:

```
AVSTAR-A: configure /site/config ab a
```

## Incorporating Configuration Changes

After testing, to put the new configuration into effect, do the following:

1. Stop any devices affected by the new configuration.
2. Take the system offline by typing:  
`offline`
3. Configure the system, using the following command format:  
`configure (master computer)`
4. Bring the system back online by typing:  
`online`
5. Wait for messages from the system being configured, and then restart the newly added devices or any devices affected by the new configuration.



*If you change the type of device on a PCU port, the entire PCU may need to be stopped and restarted.*

To list contents of the site configuration file, at the Avstar console type:

```
cat /site/config
```

The configuration file that appears on your screen is similar to the sample shown on the following pages.

Semicolons precede comments and blank lines separate sections.

The sample file (to which we refer throughout this explanation) may not match your system's file exactly, but it contains examples of the different kinds of entries you may find in your file.

```

;HOST SECTION
host    ab      a
        net      n125    20 300
        reslist   100 106 108 200 202 204 250 252 254 400
        servers   120 122 124 126 128 140 142
;
host    ab      b
        net      n125    10 30 301
        reslist   105 107 109 201 203 205 251 253 255 401
        servers   121 123 125 141
;
; ALTERNATE HOST SECTION
;
host    a      a
        net      n125    10 20 30 300 301
        reslist   100 106 108 200 202 204 250 252 254 400
        reslist   105 107 109 201 203 205 251 253 255 401
        servers   120 122 124 126 128 140 142
        servers   121 123 125 141
;
host    b      b
        net      n125    10 20 30 300 301
        reslist   100 106 108 200 202 204 250 252 254 400
        reslist   105 107 109 201 203 205 251 253 255 401
        servers   120 122 124 126 128 140 142
        servers   121 123 125 141
;
; DEVICE/RESOURCE/SERVER SECTION - BODY
;
ccu      10      ccu10      at      11 12 13 14 15 16 17 18
;
printer  11      2400-8n    2              ; Laserjet 4
terminal 12      19200      0      news      -              ; edit rm 3
terminal 13      9600-8n    1      news      -              ;
wire     14      2400      anpa7    RT      -              ; Reuters
terminal 15      9600      1      news      -              ;
dialup   16      38400-7ea  0      news      D16          ;
printer  17      9600-8n    1              ; script
wire     18      1200-8n    dummy    A2      -              ; backup AP
;
ccu      20      pcu20      pc      21 22 23 24 25 26 27 28
;
wire     21      9600-8n    dummy    DD      -              ;
terminal 22      9600-8n    1      news      -              ;
terminal 23      19200-8n  1      news      -              ;
terminal 24      19200-8n  0      news      -              ;
line     25      9600-8na   modem    hayes    -              ; dialout
dialup   26      38400-7ea  0      news      -              ;

```

```

;type      number speedprinterprogram  name
terminal   27   19200-8n    1 news      -      ;
unused     28                                     ; bad port
;
ccu        30    pcu30      pc      31 32 33 34 35 36 - -
;
line       31    9600-8n    modem    hayes    ;dialout
wire       32    9600-8n    anpa7 AP  -      ;AP wire
terminal   33    19200-8n    1 news    -      ;
terminal   34    19200-8n    1 news    -      ;
terminal   35    19200-8n    1 news    -      ;
dialup     36    38400-7ea    0 news    D36    ;
;
;
asws       200   125.1.100.85    0 gnews    -      ; News Director
asws       201   -          1 gnews    -      ; asws
asws       202   -          1 gnews    -      ; asws
asws       203   -          1 gnews    -      ; asws
asws       204   -          1 gnews    -      ; asws
asws       205   -          1 gnews    -      ; asws
asdos      250   -          2 news     -      ; andos
asdos      251   -          1 news     -      ; andos
asdos      252   -          1 news     -      ; andos
asdos      253   -          1 news     -      ; andos
asdos      254   125.1.100.83  1 news    -      ; hansen
;
terminal   100   0          1 news     -      ; net term
resource   105   console    -          ;
resource   106   archive    -          ;
resource   107   net        -          ;
special    108   0 txnet     108      -      ; txnet
special    109   0 rxnet     -          ; rxnet
;
server     120   mailserver  120      -      ; mailserver
server     121   printserver 121      -      ; slave printing
server     122   seek        122      -      ; seek server
server     123   keyword     123      kwd     ; kwdserver
server     124   action      124      -      ; timed-action
server     125   action      125      -      ; action server
server     140   monitor     140      -      ; 6 pm rundown
server     141   monitor     141      -      ; 11 pm rundown
server     142   monitor     142      -      ; show.cnn.rundown
;
mcspc      300   mcspc1    avidapdriver  apl     ; airplay
mcspc      301   mcspc2    avidapdriver  ap2     ; airplay 2
;
websession 400
websession 401

```

The site configuration file is divided into two major sections: the *host section* and the *device section* (or *body*). The host section contains information about various configurations your system can run, and devices used in each of those configurations.

The format for each host section is:

```
host <system configuration> <computer>
      <net>
      <reslist>
      <servers>
```

Parameter	Description
System configuration	Refers to whether the system is running single, dual, or triple. For instance, a standard installation with two Avstar Servers runs in a dual configuration: AB. Installations with three servers are known as triple systems, and normally run in ABC configuration.
computer	Refers to the particular Avstar Server in the system that runs in this configuration. The Avstar Servers are computers connected to the Avstar console, run the Avstar application software, and contain the Avstar database. See “Selecting Servers” on page 2-5 for more information.
net	Refers to devices configured on that particular Avstar Server in that system configuration (Communicate via the network).
reslist	Refers to resources configured on that particular Avstar Server in that system configuration.

Parameter	Description
<code>servers</code>	Refers to various utility programs called <i>servers</i> that are configured to run on that particular Avstar Server in that system configuration. This term should not be confused with the computers, also called <i>servers</i> , which run the Avstar application software. See “Servers” on page 14-1 for more information.

The *top* host section details the device, resource, and utility program numbers that run on the A server in a dual AB configuration. The *second* host section details ones assigned to server B.

Information in the second host section is used by the system if one of the servers fails. In the sample site configuration file, the `host b b` section contains a list of all the devices, including ones that normally run on server A. If A experiences a failure and is shut down, the system can be reconfigured to run all devices, resources, and servers (utility programs) on B.

When you run the `configure` console command, the master computer (usually server A) looks at the current system configuration and then assigns devices listed for each Avstar Server in that system configuration to each Avstar Server.

In the sample site configuration file, the odd numbered devices are assigned to server B and even numbered devices are assigned to server A in a dual AB configuration.

Any item number listed in the host section must have a corresponding line in the device section or body of the configuration file, and vice versa. For instance, if you are adding an Avstar Workstation resource to the body of the file, you must also add it to one or more host sections so the system knows which server would be responsible for it under various conditions.



## Changing the Configuration File

Whenever you add, remove, or modify devices on your system, you need to make corresponding changes to the configuration file—also referred to as the `/site/config` file. Changing this file requires the use of `ed`, the UNIX line editor. See “`ed`, the Line Editor” on page 10-1 for more information.

In this procedural example, a terminal is added to PCU 10, port number eight.

To edit the configuration file, do the following.

1. Select *all* servers. See “Selecting Servers” on page 2-5 for more information.
2. Type:
 

```
ed /site/config
```
3. Add device number to configuration file.

For SCO/SGI Systems:

You can use device numbers in the range 1-3000.

For instance, add the workstation’s device number to PCU 10’s configuration line. The line, at first, looks like this:

```
ccu 10 pcu10 at 11 12 13 14 15 16 17 -
```

The number 18 is determined by the formula:  $PCU\# + Port\# = Device\#$ . See “PCU Device Numbering” on page 11-34 for more information.

Change the dash (-) at the end of the line to the number 18, so the line looks like this:

```
ccu 10 pcu10 at 11 12 13 14 15 16 17 18
```

4. Add a configuration line for the new workstation to the end of this PCU definition. In this example, the last line in the definition describes device 17. A configuration line for workstation 18 is added below this line.

```
printer      17 1200-7e 1          ;[4-04]newsrm-TI880
terminal     18 9600-7e 4 news smith ;user smith
```

Although the computer is typically referred to as a workstation, it must be called a `terminal` in the configuration file.

The new line begins with the word, `terminal`, followed by the workstation’s device number, 18. The workstation communicates at 9600 baud, 7-bit, even parity. It uses printer number 4, and

the program is `news`. The workstation has a device name of `smith` to identify the person who usually uses it.

Do *not* use an uppercase (`W`) in step 5. See “ed, the Line Editor” on page 10-1 for more information.

5. Type `w` to write (save) your changes to disk.
6. Type `q` to quit `ed`.
7. Test your configuration changes.

Use the following form of the `configure` console command:

```
configure /site/config <system> <computer>
```

*System* refers to the set of servers that make up your Avstar system, while *computer* is the server whose configuration you have changed. In the previous procedural example, a workstation is added to PCU 10, which is connected to server A in system AB. To test this change, type:

```
configure /site/config ab a
```

When the prompt returns, the configuration file has been checked. If the system detects any errors, it displays `bad configuration` messages. To help you debug the file, `configure` displays the line numbers of any lines that have errors.

---

## The Hosts File (/etc/hosts)

The hosts file (/etc/hosts) is a road map to other computers on the network that your Avstar Servers need to know about or communicate with. It lists IP addresses of other computers and the computers' names, along with any alternate names or aliases by which the other computers are known. Workstations need not be listed in the hosts file, but putting them in will maintain an inventory of already used IP addresses. If a computer is not listed in the hosts file, the Avstar Servers will not know about it. This includes PCU units, archive computers, routers, and mail gateways.

## System Profile Files

For your system's servers to work together, they need access to basic system information, such as how many computers are in the system and how the computers are connected. Each server must also have access to system parameters, such as the default read rate and script margins. This information is kept in *system profile files* whose names reflect the information they contain. For instance, information about wires is kept in /site/wires and information about printers is kept in /site/printers.

Sample system profile files are in Appendix B. The most important of these profile files is the site profile file (/site/system), which is discussed in the sections that follow. When you start your system, each server reads its copy of the system profile and incorporates the material into its operation.

## Viewing the System Profile File

The system profile is a text file stored in `/site/system`. Because it is a text file, use the `cat` command to list its contents.

For instance, when you type `cat /site/system`, the system output looks similar to the following:

```
AVSTAR-A:cat /site/system
id=AVSTAR net=ab
textmax=80          scriptlhmax=20      scriptrhmax=20
airmargin=20        airshift=yes
lowwater=20         highwater=25      purgelimit=5
timechar=:          outtime=30:00     readrate=180
localtimeout=180:00 remotetimeout=0:00 pausetimeout=0:05
logintimeout=1:00   netlogintimeout=0:00
min_passwd_length=5 maxhits=250        timing_template=200
security=or         load=5
auto_upgrade=yes
```

The system profile contains several parameters. Each parameter begins with an identifying keyword, followed by an equal sign (=) and the parameter's value. When an Avstar Server reads its system profile, it finds each parameter's value by searching for the keyword that represents that parameter.

For instance, to find the system ID, each server searches its system profile for the keyword, `id`, and reads the parameter associated with that keyword. If the server searched the system profile in the example above, it would find that AVSTAR was the parameter associated with that keyword.

Most, but not all, parameters have default values the system uses if the parameter is not present in the system profile. Consequently, a system profile usually includes only parameters that you want to set differently from their default values, and those that have no default values and must be set in the system profile. Your system profile may not contain the same parameters as the example.

## Changing the System Profile File

Use the `ed` console command to modify the system profile in `/site/system`. See “ed, the Line Editor” on page 10-1 for more information. Then shut down and restart the system to get the servers to read the modified system profile. System profile parameters are configured in the system when servers are connected.

For instance, to change the `localtimeout` parameter in your system profile file, do the following at the Avstar console:

This procedure, which modifies the `/site/system` file uses `ed`, the UNIX line editor. If you do not know how to use `ed` to modify lines in the file, please see “Using the UNIX Line Editor” on page 10-4.

1. Select *all* servers. See “Selecting Servers” on page 2-5 for more information.

2. Type:

```
ed /site/system
```

A message similar to the following appears:

```
editing /site/system
213
```



*All files in the Site directory, including the system profile and site configuration file, can contain only Roman characters.*

3. Find the line that contains the `localtimeout` parameter, such as:

```
scriptrhmax=20 localtimeout=45:00 remotetimeout=30:00
```

4. Change the `45:00` value for that parameter to `15:00`:

```
scriptrhmax=20 localtimeout=15:00 remotetimeout=30:00
```

5. Type `w` to write the change to disk.

6. Type `q` to quit.

Do *not* use an uppercase (`w`) in step 4. See “ed, the Line Editor” on page 10-1 for more information.



*When you modify your system profile, separate parameters from each other with spaces, tab spaces, or carriage returns.*

7. Shut down the system, and then start it up again.

When you start the system, each server reads its system profile and incorporates parameters in that file in its operation.

## Listing Parameter Settings

To find out which parameters the system is using, type the **status all** console command. This command displays the system profile settings the system has incorporated in its operation.

Information similar to the following appears:

```
A is ONLINE and has been CONFIGURED. ID is AVSTAR.
System is AB. Master is A.
Disk status is OK
System was last configured at Wed Aug 30 17:54:04 2000

airmargin=40          logintimeout=01:00    remotetimeout=00:00
airshift=yes         maxhits=500          scripthmax=40
auto_upgrade=yes     min_passwd_length=3  scriptrhmax=40
clockmax=12          netlogintimeout=00:00 security=OR
excludedvideo=none   outtime=30:00        textmax=80
lastlogin=yes        pausetimeout=00:05   timechar=:
load=0               readrate=180         timing_template=200
localtimeout=15:00
```

The **status all** console command lists values for all parameters defined in the profile, except for the low and high watermarks and the purge limit. Parameters not explicitly defined in the system profile do not appear in this list.

To list current low and high watermarks and the purge limit, type:  
**cat /site/system**

A low watermark set to 20 would equate to 5000 database blocks (20x250=5000). A high watermark set to 25 would equate to 6250 database blocks (25x250=6250). See “Monitoring Free Space” on page 19-2 for more information.

## System Profile Parameters

Each system profile description below includes information about the values that parameter can have and whether or not the parameter has a default value. If there is a default value, it is underscored. If a parameter does not have a default value, you must give it a value in the system profile.



*The connect command can override anything in the profile by adding <parameter>=<value> on the connect command line.*

`airmargin=<number of columns> (25)`

Determines how many characters wide a script is when displayed on a teleprompter or workstation using the AIR or AIRPROMPTER command. The value should be the same as that set for the `scriptrhmax` option in your system's printer profiles. The `scriptrhmax` in a printer profile is not the same as `scriptrhmax` in the system profile. The default value is 25, the minimum value is 1, and the maximum value of 40 is not enforced.

`airshift=[yes | no]`

If set to `yes`, your system shifts all text to uppercase before sending it to the teleprompter. If set to `no`, it does not change the text of a story before sending the story to the teleprompter. The default is `airshift=yes`.

`auto_upgrade=[yes | no]`

Determines whether a user running an outdated version of the Avstar Windows-based Workstation (client) software is allowed to upgrade it automatically. If set to `no`, it means users of outdated software are not asked if they want to upgrade. The default is `auto_upgrade=yes`.

`clockmax=[12 | 24]`

Determines how backtimes/cumetimes (cumulative) are displayed. This setting only applies to VT sessions, which can display

the time in 12 or 24-hour format. Avstar Workstations (ASWS) always displays these times in 24-hour format.

`excludedvideo=[director | none]`

Determines the handling of director video when received in Story Exchange Protocol (SEP) format or when saved in a story from a VT workstation. If set to `none`, this text is converted into bold italic text. If set to `director`, the text is converted into closed caption (CC) text.

`highwater=<# of 250 block units> (10)`

Establishes the upper limit to which `dbserver` attempts to rebuild the free list. Set this parameter far enough above the low watermark so the system is not in danger of slipping beneath that mark. The number you enter represents units of 250 blocks of database space, where a block is one kilobyte. For instance, a high watermark of 25 (recommended for most systems) represents 6250 blocks. The default value is 10.

`id=<system name>`

Gives the system a name. The system uses this name in some of its messages and in the prompts. The system name defined here must match that used in the `/etc/hosts` file (a UNIX file used by networking software), must be in uppercase, and can be up to eight characters long. There is no default value, so you must set it in the system profile.

`lastlogin=[yes | no]`

Lets you suppress on a user's workstation display of last time user was logged in. Setting `no` accomplishes this; the default is `lastlogin=yes`.

`load=<number of connections> (0)`

Specifies the maximum numerical difference the system tries to maintain between network connections from Windows-based and DOS-based clients on different servers in your system. This is called *load balancing*, and it is intended to keep one server from



handling a much higher number of connections than any other server.

For instance, if you set this parameter to 5, the system distributes connection requests so the difference in the number of connections is no higher than 5. If you had 2 active connections on server A, and 7 on B, the next request to connect to B would be shifted to A. Connection requests for A would be allowed, until the number of connections on A was 5 more than on B.

The default value for this parameter is 0, which means load balancing does not occur.

```
localtimeout=<min:sec> (00:00)
```

As a security precaution, your system automatically logs out workstations if they are idle longer than the time set in this parameter. This applies to all workstation types except dialup workstations, which are subject to the `remotetimeout` value.

If a story is open at an idle workstation, the system saves the story before logging out the workstation. Setting `localtimeout=00:00` prevents the system from logging out workstations. This is the default setting if it is not set in the system profile.

The `localtimeout` parameter is set in minutes and seconds. For instance, to have the system log out any workstation idle for more than two hours, set this parameter to `120:00`. The maximum value is 540 minutes.

```
logintimeout=<minutes>:<seconds> :60
```

Sets the time a user has to log in when using a dialup workstation. A modem control signal is dropped (disconnecting the modem) if the time limit is reached without a successful login. The maximum value is 32,400 seconds (540 minutes). If you set this parameter to 0 or omit it, your system uses the default value of one minute or 60 seconds.

```
lowwater=<# of 250 block units> (5)
```

Establishes minimum disk space that the system tries to keep available for immediate use. Use it with the `highwater` and

`purgelimit` parameters to control how the system recycles space in the database. Set this parameter in units of 250 database blocks, where a block is one kilobyte (1KB). For instance, a low watermark of 20 (recommended for most systems) equals 5000 blocks of database space.

If the number of blocks in the free list falls below the low watermark, the system runs `dbserver` to reclaim the oldest stories from the Dead queue, recycling the space onto the free list. This continues until the free list is restored to the high watermark. If you do not include this parameter, the system uses the default value of 5 (1250 blocks).

`master=<computer name>`

Designates one of your system's servers as the master computer, which controls all database activity and performs the majority of housekeeping, such as running `dbpurge` every hour and invoking `dbserver` when low on space.

Generally, this parameter is left out of the system profile, causing the system to designate as the master computer the server whose name is alphabetically first (usually server A). You can specify a server as the master computer using the `connect` console command. The format is:

`connect <computer> master=<name>`

`maxhits=<number of hits> (500)`

Defines maximum number of *hits*, or matches, that a background search—including Fast Text Search (FTS)—can find. For instance, setting this parameter to 50 limits the total number of hits in a single search to 50. The maximum number you can specify is 32765. If you do not assign a value, the system uses a default value of 500.

`min_passwd_length=<number of characters> (5)`

Defines minimum password length for your users. For instance, setting `min_passwd_length=6` prevents users from creating passwords shorter than six characters. The value may range between 1 and 12 characters. This does not apply to passwords

you assign with `utraits`, but if you assign a password that is too short, the user will be forced to change it the next time he or she logs in. The system uses 5 as the default value.

```
msgserver=[silent | verbose]
```

Used only for debugging. To find out whether or not a mailbox is working, set this parameter to `verbose`. This causes the console to display a message whenever activity occurs in a queue with a mailbox assigned to it. Change this parameter while the system is running using the `msgdebug` command. The default setting is `silent`. This prevents messages regarding mailbox activity from being displayed.

```
name=<computer name>
```

Each server must have a unique name (either A, B, C, or D) to distinguish it from the other servers in the system. Typically, assign these names during the startup process using the `connect` console command—`connect a` for server A, and so forth—so it need not appear in the system profile. There is no default value assigned.

```
net=<computer names>
```

If your system's servers are connected over an Ethernet network, include this parameter in the system profile. This allows you to specify all servers in the network. For instance, in a system using three servers named A, B, and C, this parameter would be set to `net=abc` in the system profile.

Include this parameter only if your servers are connected on a network. Using this parameter precludes use of the `single` parameter. There is no default value assigned.

```
netlogintimeout=<min:sec> (1:00)
```

Sets time allowed for a user to log in to the network. The connection is terminated if the time limit is reached without a successful login. The maximum value for this parameter is 32400 seconds (540 minutes). If set to 0, the automatic time-out is disabled. If not

set, your system uses the default value of 0:00. This profile option only applies to Avstar DOS logins.

```
outtime=<min:sec> or <hr:min:sec> (60:00)
```

Sets default show ending time (out time) in minutes and seconds—or the time of day the show will end in hours, minutes, and seconds. If the parameter is in the `min:sec` format, the system uses that value as the default show length.

Hours display in the 12-hour clock format, so the value 12:00:00 could be either noon or midnight. Actual time values can be between 01:00:00 and 12:59:59.

If you do not set an out time, the system profile value is used to backtime a rundown. The maximum time value is 540:00 (9 hours). If not set, the system assumes a default out time of 60 minutes.

```
pausetimeout=<min:sec> (00:30)
```

Sets a default value for the PAUSE command, which is used in some keyboard definitions. Users can override this default value. If not set, the system assumes a value of 30 seconds.

```
purgelimit=<hours> (0)
```

If `dbserver` reclaims all space available in Dead queue without restoring the free list to the low watermark, it begins to purge old stories by making a series of passes through the database. On each pass, `dbserver` temporarily decreases each queue's purge interval by one hour and removes any stories older than the new purge interval. It continues doing this until it has rebuilt the free list to the high watermark or reaches the purge limit.

The purge limit sets the maximum number of passes `dbserver` can make through each queue. The total number of hours `dbserver` can purge from a queue is equal to the queue's purge interval minus your system's purge limit.

Use the purge limit to prevent `dbserver` from purging everything from important queues in its attempt to build up the free list. For instance, if you set the purge limit to two hours, queues with a

3-hour purge interval retain at least one hour's worth of stories, even in a low-on-space situation.

You can set the purge limit between 0 and 24. If not set, your system uses a default value of zero hours, which prevents `dbserver` from purging any queue beyond its purge interval.

```
readrate=<words/minute> (180)
```

Sets the system's default read rate. When you add a user to the system, the Add New User dialog box will default to a read-rate of zero, which will be replaced by the `/site/system` read rate when needed. After a user has been added, you can change the user's read rate using the Modify User Account dialog box. See "Modifying User Traits" on page 4-1 for more information. If not set, the system uses a default read rate of 180 words per minute.

```
remotetimeout=<min:sec> (00:00)
```

This parameter sets the time a dialup (remote) workstation can be idle before the system logs it out. This time-out value also applies to all connect sessions, including sessions that connect a workstation at your station to another service. If a story is open when the system logs out the workstation, the story is saved. Disable the automatic logout of idle remote workstations and connect sessions by setting this parameter to `00:00`. The maximum value is `540:00`. The default that the system uses is `00:00` if this parameter is not included in the system profile.

```
scriptlhmax=<number of columns> (40)
```

Sets default width of left column of a scripted story. The width is set as a number of characters from the left side of the screen. For instance, a value of 20 causes the left column of a scripted story to be 20 characters wide, beginning from the left edge of the workstation screen. Users can override this setting using width parameters with the `SCRIPT` command. The allowable range for this value is 2 to 78, inclusive. If not set, the system uses a default value of 40.

`scriptrhmax=<number of columns> (40)`

Sets default width of right column of a scripted story. The width is set as a number of characters from the end of the left column. For instance, a value of 20 causes the right column of a scripted story to be 20 characters wide, beginning from the end of the left column. Users can override this setting using width parameters with the `SCRIPT` command. This is not the same as `scriptrhmax` in the printer profile. The allowable range is 2 to 78, inclusive. The sum of the `scriptlhmax` and `scriptrhmax` values must not be greater than 80.

If not set, the system uses a default of 40. If you use an electronic teleprompter, set this parameter to a value one greater than the value to which the `airmargin` parameter is set. This ensures that scripts on the teleprompter are displayed as they appear when aired on the workstation.

`security=[and | or]`

Indicates how your system determines group access of a particular user with a particular workstation (or other device). If set to `and`, both user and workstation must be members of the same group for the user to gain access to directories or queues assigned to that group. If set to `or`, the user can access any database items assigned to groups containing either the user or workstation. The default value is `or`.

`single=<computer name>`

Tells the system it is running on only one server and names that server. If your system consists of a single server, include this parameter. Generally, in systems with only one server, the name is `A`, and this parameter is set to `single=a`.

`textmax=<number of columns> (80)`

Determines how many characters wide a story will be when a user executes the `SCRIPT UNDO` command. The user can override this value with `PACK` or `SCRIPT UNDO` commands.

If you do not include this parameter in the system profile, the system uses the default value of 80 characters (also the maximum value). Generally, this parameter is not included in the system profile, and the system uses the default value.

```
timechar=<character> (:)
```

Defines the character the system uses to separate hours, minutes, and seconds in time displays. For instance, using a colon as the time character displays the time as hh:mm:ss. A colon is the default.

```
timer=[silent | verbose]
```

Your system contains a timer program that is always running. If this parameter is set to `verbose`, the server sends a time display to the console every 15 minutes.

```
A Sat Apr 1 14:45:00 2000 Avstar News
```

`verbose` is the default setting, so do not include when you want timer messages to appear. To disable time displays, include this parameter as `timer=silent`.



**All time values in the system profile must be set in minutes and seconds in the format <mm>:<ss>. Set a value for both minutes and seconds. For instance, to specify a time of two minutes, type: 2:00. To specify a time of 25 seconds, type: 0:25. (The out-time value can also be set as a time of day in the format <hh>:<mm>:<ss>.)**

## Adding Devices to Your Avstar System

When you add a device, such as a new printer, to your Avstar system, you need to put information about it in the appropriate configuration file(s). This section gives you information about how to do that. It includes complete procedures for adding PCUs, workstations, printers, and wires. For information about adding connect services, see Chapter 17, *Connect Services*.

## Adding a PCU

This section contains information about adding a Peripheral Controller Unit (PCU) to your system. For a definition and other information about PCUs, see Appendix D.

To add a PCU to your system, do the following:

1. Choose a device number for the PCU.

By convention, PCU device numbers are multiples of 10. Choose the next available multiple of 10. If the system already has four PCUs, numbered 10 through 40, use 50 as the device number for the new PCU.

2. Connect PCU to network.
3. Add PCU to appropriate host definitions in the configuration file—that is `/site/config`—using the UNIX line editor. See “ed, the Line Editor” on page 10-1 for more information.

The net line in each server’s host definition begins with the word `net` followed by a hyphen or a `<network name>` defined in the `/etc/networks` file. Follow this with a list of network PCUs that you want to run on that computer.

For instance, in the host definitions shown below, PCUs 10 and 20 are network PCUs running on server A, while PCUs 30 and 40 are network CCUs running on server B.

```
host ab a
      net - 10 20
;
host ab b
      net - 30 40
```

- a. Add new PCU to server B, so its host definition looks like this:

```
host ab b
      net - 30 40 50
```

- b. Add new PCU to the appropriate alternate host definitions.

4. Add a PCU line to the configuration file `/site/config`.



A PCU configuration line must begin with the word `ccu` followed by the PCU's device number, PCU name, and type. These are followed by a list of devices connected to the PCU. The format for a PCU configuration line is:

```
ccu <device #> <name> <type> <dev 1> <dev 2> ... <dev 8>
```

Parameter	Description
<code>device #</code>	The PCU's device number. It must be listed in a host definition. By convention, this is a multiple of 10.
<code>name</code>	The PCU's name. Can be any string, as long as it begins with a letter and is no longer than six characters. PCU names that appear in the configuration file must match those listed in your <code>/etc/hosts</code> file.
<code>type</code>	PCU type is <code>pc</code> .

End the configuration line with a list of devices connected to the PCU, with their numbers separated by spaces. The order in which you list devices must correspond to the PCU ports to which they are connected—the device connected to port 1 must be first, the device connected to port 2 must be second, and so on.

You can include up to eight devices in a PCU configuration line. By convention, the device number is the sum of the port number and the PCU's device number. See “PCU Device Numbering” on page 11-34 for more information. Fill unused ports with hyphens (-).

Add the line following the device lines for PCU 40. Because devices are not connected to the new PCU, the line would look like this:

```
ccu 50 pcu50 pc - - - - - - - -
```

5. Add the PCU to the hosts file `/etc/hosts`.

A PCU's name must be a valid Internet name. It must resolve to an Internet address, either by being included in the hosts file or by an Internet name server.

An example PCU line from the hosts file may look like this:

```
125.0.10.50 pcu50    02608c301052
```

The hosts file also contains lines identifying each server in your system.

6. Add the PCU to the `/etc/bootptab` file.

You must list each PCU running on your system in the file `/etc/bootptab`. You may need to create this file if you are adding the first PCU to the system. Format of a line in this file is:

```
<name> <type> <Ethernet address> <Internet address>
<bootfile>
```

Parameter	Description
name	Put PCU's name, such as <code>pcu50</code> , in this position.
type	Always put one (1) in this position.
Ethernet address	The PCU's Ethernet address is found on its Ethernet card.
Internet address	The PCU's Internet address is assigned when the PCU is installed. This address must match the one listed in the <code>/etc/hosts</code> file for this PCU.
bootfile	Insert <code>pcuos.exe</code> .

A typical line in `/etc/bootptab` would look like this:

```
pcu50 1 08002b33a077 125.1.0.33 pcuos.exe
```

SGI Irix Systems

For SGI Irix systems, the path and default bootfile must exist for the bootp server to work correctly. Put the `/exc/ccu/at` and `pcuos.exe` files in `bootptab` for SGI systems. Alternately, you

could create an empty file `defaultboot` directory, `/usr/local/boot`. See the software installation information that came with your system.

If you are connecting a PCU to a system running SCO UNIX, the format of lines in `/site/config` and `/etc/hosts` are identical to those described above. However, `/etc/bootptab` uses the following format:

```
<PCU name>: ht=1: ha=<Ethernet address>:
ip=<internet address>:
```

Parameter	Description
PCU Name	Host name of PCU, such as PCU50, in this position.
Ethernet address	PCU's Ethernet address is found on its Ethernet card. (Do not use colons in this address on SCO UNIX systems, since the colon character is used to separate parameters on the line.)
internet address	PCU's Internet address is assigned when PCU is installed. Address must match one listed in <code>/etc/hosts</code> file for the PCU.

A typical CCU line in the `/etc/bootptab` file for a SCO UNIX system would look like this:

```
pcu50 ht=1: ha=08002B33A077: ip=125.1.0.33:
```



*The colons after each parameter are required.*

7. (Optional) Use the `configure` command to test your changes.

Use the following form of the `configure` command:

```
configure /site/config <system> <computer>
```

In the previous example, PCU 50 is added and running on server B in an AB system. To test this change, type:

```
configure /site/config ab b
```

When the prompt returns, the configuration file has been checked. If the system detects any errors, it displays bad configuration messages.

8. Select the master computer (typically server A). See “Selecting Servers” on page 2-5 for more information.
9. Become a console superuser. See “Becoming a Console Superuser” on page 3-2 for more information.
10. Take the server (master computer) offline.

```
AVSTAR-A# offline
```

11. Type the configure command at the prompt.

```
AVSTAR-A# configure
```

12. Bring the server back online again.

```
AVSTAR-A# online
```

13. Exit superuser mode. (CTRL-D)

A message similar to the following will appear:

```
A Wed Oct 4 00:18:58 2000 msg System being configured
```

14. Start the new PCU.

```
AVSTAR-A: restart 50
```

15. (Optional) Back up your site files with the `sitedump` command.

## PCU Device Numbering

Each PCU and device connected to it must have a unique device number. Use device numbers to refer to devices when using commands such as `restart` and when adding configuration lines to your configuration file. Your system uses device numbers in its console messages, such as `failed to load device 11`.

You can randomly assign different numbers to devices, but system maintenance is simpler if you give each device a number that corresponds to the PCU port to which it is connected. The numbering con-

vention described here has been established to make your system maintenance work easier.

Network PCUs are usually given device numbers that are multiples of 10. These numbers are usually sequential and begin with the number 10. So, in a system with four network PCUs, the PCUs would have 10, 20, 30, and 40 as device numbers.

When your system was installed, each device connected to a PCU was given a number that is the sum of the PCU's device number and the number of the PCU port to which the device is connected. For instance, you would give a workstation connected to port 1 on PCU 10 device number 11. A printer connected to port 5 on PCU 20 would have device number 25.

Numbering your system's devices in this way lets you quickly determine which PCU—and which port on the PCU—a device is connected to just by looking at the device's number.

## Adding a Workstation

To add any type of new workstation to your system:

1. Choose a device number for the workstation.
  - For a serial workstation or dialup line, find an unoccupied PCU port.
  - For a network device, determine the next available number in the range you have set aside for these devices.
2. Connect workstation to PCU or network.
3. Add workstation to configuration file stored on each server in your system.

If you are adding a network workstation or a DOS PC, add the workstation's device number to a `reslist` line in the host definitions. For serial devices, be sure the workstation (or dialup) is listed in the configuration line of the PCU to which it is connected.

Every type of workstation needs its own configuration line. To configure a workstation, use the format:

```
<platform> <dev #> <speed|addr.> <printer #>
<prog.> <dev name>
```

Parameter	Description
platform	The type of workstation defined: asws for Avstar Workstation terminal for VT terminal asdos for DOS PC workstation dialup for dialup modems
device #	Workstation's device number.
speed/ address	For terminal and dialup workstations, the speed at which the workstation communicates. Also indicates bits per character, parity, and handshaking. For instance, 7e is 7 bits per character and even parity, 8n is 8 bits per character and no parity. For a network workstation, set the speed to zero.  For Avstar and DOS workstations, if you use a hyphen (-), this resource is available to any licensed PC on your system. If you place an IP address here, this resource can be used only by the PC with that address. Additionally, this can be the Ethernet address assigned to the PC's network card.
printer #	Default printer. This number must be assigned to a printer listed in the configuration file. Entering 0 has the workstation print to a printer connected directly to the workstation.
program	Program that runs the workstation. For Avstar Workstations, use gnews. Otherwise, this is always news.

Parameter	Description
<code>device name</code>	<p>A device name up to eight characters used for group security. If you assign a device name to a workstation, you can grant security permissions to the workstation by adding that name to the appropriate group story.</p> <p>When someone edits and saves a story at this workstation, its device name is placed in the <code>devname</code> field (if one exists) in the Story Form panel. If you do not want to give the workstation a device name, place a hyphen (-) in this position.</p>

4. (Optional) Use the `configure` console command to test your configuration changes. The syntax is:

```
configure /site/config <system> <computer>
```

For instance, type:

```
configure /site/config ab a
```

When prompt returns, the configuration file has been checked. If the system detects any errors, it displays appropriate bad configuration messages.

5. Reconfigure the system.

- a. Use `broadcast` to tell users to log out. For instance:

```
AVSTAR-A: broadcast all users please logout
```

- b. After users log out, use `list s` to ensure they are logged out.

- c. If not, you can force a log out of the PCU by using the `logout` command in this format:

```
logout <PCU number>
```

For instance, to log out PCU 10, type: `logout 10`

- d. Stop the PCU. Use the `stop` command in this format:

```
stop <PCU number>
```

For instance, to stop PCU 10, type: `stop 10`

- e. Select the master computer (typically server A). See “Selecting Servers” on page 2-5 for more information.
- f. Become a superuser. See “Becoming a Console Superuser” on page 3-2 for more information.
- g. Type:  
AVSTAR-A# **offline**
- h. Type:  
AVSTAR-A# **configure**
- i. When the prompt returns, bring the system online again by typing:  
AVSTAR-A# **online**
- j. Exit from superuser mode. (CTRL-D)

A message similar to the following will appear:

```
A Wed Oct 4 00:18:58 2000 msg System being configured
```

6. Select the appropriate server.
7. Restart the PCU to which the new device is connected, using the following format:  
  
`restart <PCU number>`  
  
For instance, to restart PCU 10, type: **restart 10**  
  
You should see `Hot-to-go` messages for devices on that PCU, including the new workstation, indicating they are ready to use.
8. (Optional) Back up site files with the `sitedump` command any time you add a device. See “Backing up Site Files” on page 20-25 for more information.



## Adding a DOS PC Workstation

To add a DOS PC workstation to your Avstar system, you must complete three phases in the setup.

- Phase 1 - Complete general configuration instructions for adding a workstation to the Avstar system.
- Phase 2 - Set up and/or confirm IP address information in the System directory.
- Phase 3 - Set up the session file on the DOS PC.

### Phase 1

The procedure for Phase 1 was explained in “Adding a Workstation” on page 11-35. Phases 2 and 3 are explained below.

### Phase 2

1. Check the number of DOS PC client resources being used on your system as defined in the `/site/config` file.

The number of people that can connect to your Avstar system with DOS PCs at any one time is equal to the number of PC client resources you set up. Your system’s license indicates the maximum number of these resources you can configure and the maximum number of users who can connect simultaneously.

If you have a site license, you do not need to put any addresses in `SYSTEM.CLIENT.DOS`. The system will accept connections from any PC.

2. If you have a site license, and you put some addresses in `SYSTEM.CLIENT.DOS`, then Avstar NRCS allows connections only from DOS clients whose Ethernet or IP addresses are listed in the `SYSTEM.CLIENT.DOS` queue. Go to an Avstar Workstation, and do the following:
  - a. Create this queue now if it does not exist. See “Adding a Directory or Queue” on page 5-3 for more information.
  - b. Verify IP addresses of all suitable PCs at your site are listed in stories in those queues.



List all your PCs in one story, or use several stories, which can contain only one address or address-range per line. The number of addresses allowed is limited to the number of licensed DOS machines—that is, if you do not have a site license, you can only put the licensed number of DOS client addresses (IP or Ethernet) in `SYSTEM.CLIENT.DOS`.

An excerpt from a typical address list story looks like this:

```
;Control Room
152.165.104.20 ; J. Arlin
152.165.187.10 ; K. Snow
152.165.111.1-152.165.111.255 ; IP Address Range
...
;Newsroom
152.165.117.73 ; P. Miner
152.165.194.38 ; S. Katz
```

You must take the system offline prior to running the console command:  
`configure -n`

3. At the Avstar console, type **configure -n** to make any changes to the `SYSTEM.CLIENT.DOS` queue available to the system.

### Phase 3

Each DOS PC must have at least one session file on it that lists the Avstar host server(s) to which it can connect. This file should be located on the DOS PC workstation in the same directory as the application `ns.exe` (typically `NS`).

To set up the DOS PC's session file, do the following at the DOS PC workstation:

1. Navigate to the `NS` directory (or the directory containing the `ns.exe` application). The DOS prompt will look similar to this:  
`C:\NS>`
2. Create a session file to connect to the Avstar system by typing **edit** followed by the host server name(s) with which the session is to be established.

For instance, at the DOS prompt, type:

```
C:\NS> edit avstar.ses
```

The DOS editor, which appears as a blue window, opens.

3. A session file can specify up to four different host server names, as defined in the `/etc/hosts` file on the PC or by DNS. The format is as follows:

```
machine <host 1> <host 2> <host 3> <host 4>
```

For instance, for a two server system, type:

```
machine avstar-a avstar-b
```



*You can list the systems on multiple lines in the file as long as each line begins with the word “machine.” You can specify host computers by Internet address rather than by name.*

## Adding a Printer

You can connect any kind of serial printer to the Avstar system, and configure each printer to take advantage of its special features.



*For more information about printers, see Chapter 12, Printers.*

Even if you never add a printer to your system, this information can help you if you need to move a printer to a different PCU or modify a printer’s profile.

To add a printer to your system, do the following:

1. Choose a device number for the printer. To do this, find an available PCU port to which you want to connect the printer. For instance, if we add a printer, such as HP LaserJet, to port 7 on PCU 10, we give the printer device number 17. See “PCU Device Numbering” on page 11-34 for more information.
2. Connect printer to PCU.

3. Create a printer queue in the database in the `SYSTEM.PRINTERS` directory. If you do not complete this step, any print requests to the new printer will not be processed. This is done at an Avstar Workstation. See “Adding a Directory or Queue” on page 5-3 for more information.

Here’s an example: The new printer is printer number 4, so the new queue is called `004`. The entire pathname is `SYSTEM.PRINTERS.004`.



*Your system can handle up to 500 printers. Print queues for the first 250 printers are in the directory `SYSTEM.PRINTERS`. Print queues for the next 250 printers are in `SYSTEM.PRINTERS2`.*

4. Return to the console to finish adding the printer.
5. Add the printer to the configuration file, `/site/config`, on each Avstar Server in your system.

This involves adding the printer’s device number to the PCU’s configuration line and adding a device configuration line for the printer.



*If you move a printer to another PCU, you must remove the printer’s device number from the old PCU’s configuration line and add it to the new PCU’s configuration line. Give the printer a new device number that reflects the new PCU port to which you attached it.*

This procedure, which modifies the `/site/config` file uses `ed`, the UNIX line editor. If you do not know how to use `ed` to modify lines in the file as required in step 5, please see “Using the UNIX Line Editor” on page 10-4.

- a. Select *all* servers. See “Selecting Servers” on page 2-5 for more information.
- b. Type:
 

```
AVSTAR-A: ed /site/config
editing /site/config
1259
```
- c. Add printer’s device number to list of device numbers in the PCU configuration line. For instance, for a printer connected to port 7 on PCU 10, the printer’s device number (17) is added

to PCU 10's configuration line. At first, the line may appear similar to this:

```
ccu 10 ccu10 at 11 12 13 14 15 16 - -
```

Ensure that you place the new device number in the position that represents the port to which you connected the printer. For instance—for the printer with device number 17—the printer's device number is placed in position 7 in PCU 10's configuration line. The new line should look like this:

```
ccu 10 ccu10 at 11 12 13 14 15 16 17 -
```

- d. Add a configuration line for the new printer. By convention, configuration lines are arranged according to device number. For instance, the configuration line for the printer with device number 17 would be placed after the configuration line for device 16.

A regular printer configuration line uses the format:

```
printer <device #> <speed> <printer #>
```

Parameter	Description
device #	The printer's device number. It identifies the printer to which the configuration line applies. Include this number in the appropriate PCU configuration line.
speed	Sets the baud rate (bits per second) at which the printer communicates with the PCU.
printer #	Assigns a logical printer number to the printer. Used in workstation and dialup configuration lines to identify the printer that devices use as their default printer. This is not the same as the printer's device number, but it is the same number that you use to refer to the printer when you print to it from an Avstar Workstation.

For instance, the printer uses device number 17 and communicates at 1200 baud, 7-bit, even parity, and users will refer to this printer as printer 4. So, the new printer line looks like this:

```
printer 17 1200-7e 4 ;newsroom HP LaserJet
```

Do *not* use an uppercase (w) in step 4. See “ed, the Line Editor” on page 10-1 for more information.

6. When you finish making changes to the configuration file, you must save your changes. To do this, type:

w

After you press Enter, a number will appear representing the file size, such as:

1306

7. Exit the UNIX line editor by typing:

q

8. Create a profile for the printer.

The easiest way to create a profile is to copy the standard profile for the type of printer you are adding to a file for the new printer’s profile. Then make any necessary modifications to the copy you just made.

A printer’s profile must be in a file with the same name as the printer’s device number. Also, this file must be located in the `/site/printers` directory.

For instance—for an HP LaserJet printer connected to port 7 on PCU 10—you would copy the standard profile for the HP LaserJet (`/site/printers/hplaser`) to a new file called `/site/printers/17`. Then modify the profile so the system can use a font module that has been added to the printer.

To copy a profile to a new file, do the following:

- a. Select *all* servers. See “Selecting Servers” on page 2-5 for more information.



**As with all site files, each of the system’s servers must have its own copy of the printer’s profile, so ensure that you select *all* servers before you copy the standard profile.**

- b. Use the `cp` command to copy the standard profile, such as `/site/printers/hplaser` for an HP LaserJet printer into a new file on each server, such as `/site/printers/17`.

The format for this command is:

```
cp <file pathname copied> <new file pathname>
```

The entire command line would look similar to this:

```
cp /site/printers/hplaser /site/printers/17
```

9. Use the `cat` command to examine the file and determine whether you need to make any changes or additions. The format is:

```
cat <file pathname>
```

For instance, to examine the file for the printer connected to port 7 on PCU 10, type:

```
AVSTAR-A: cat /site/printers/17
```

A message similar to the follow will appear:

```
; HP LaserJet IVsi Printer profile
;
ejectcode          <ff>
ejectcount         1
idlecount          0
pagelength         66
scriptrhstart      32
scriptrhmax        23
scriptlhmax        25
scriptshift        yes
scripttemplate     no
;
expand    <esc>(OU<esc>(sp7.0h18vs0b11T<nul>    ;Courier
;
font 2    <esc>(s3B          <esc>(s0B          ;Bold
font 3    <esc>(s4B          <esc>(s0B          ;Extra Bold
font 4    <esc>&dD          <esc>&d@          ;Underline
font 5    <esc>&dD<esc>(s3B<esc>&d@<esc>(s0B      ;Bold/Under
;
form 7    <esc>&100<esc>(s10H<esc>(s12V <nul>    ;landscape
form 8    <esc>&11H          <nul>            ;top tray
form 9    <esc>&14H          <nul>            ;lower tray
```

10. (Optional) Test your configuration changes by using:

```
configure /site/config <system> <computer>
```

For instance, a printer is added to PCU 10, which is connected to server A in an AB system. To test this change, type:

```
configure /site/config ab a
```

When the prompt returns, the configuration file has been checked. If the system detects any errors, it displays `bad configuration` messages.

11. Do one of the following:

- a. If you modified *only* an existing printer's profile, you need to restart the printer to incorporate the changes. To do this:

1. Select the server to which the printer's PCU is connected.
2. Use the `restart` command in this format:

```
restart <device number>
```

For instance, to restart the device at port 7 of PCU 10, type:

```
AVSTAR-A: restart 17
```

```
AVSTAR-A: P17: 12:43:20 Hot-to-go
```

When you see a `Hot-to-go` message for the printer, as shown in the previous example, you can begin to use it. If there are any errors in the profile, the system reports them on the console when you restart the printer.

-OR-

- b. To have the system adopt changes you have made to the configuration file, reconfigure it by doing the following:

1. Log out the PCU. For instance, to logout PCU 10, type:

```
logout 10
```

2. Stop the PCU. For instance, to stop PCU 10, type:

```
stop 10
```



3. Select the master computer (typically server A). See “Selecting Servers” on page 2-5 for more information.
4. Become a console superuser. See “Becoming a Console Superuser” on page 3-2 for more information.
5. Take the system offline. Type:  
AVSTAR-A# **offline**
6. Reconfigure the system. Type:  
AVSTAR-A# **configure**
7. When the prompt returns, use `online` to bring the system back online again. Type:  
AVSTAR-A# **online**
8. Exit from superuser mode. (CTRL-D)

A message similar to the following will appear:

```
A Wed Oct 4 00:18:58 2000 msg System being configured
```

9. When you see the `System being configured` message, select the server to which the PCU is connected and use the `restart` command. For instance, to restart PCU 10, type:

```
AVSTAR-A: restart 10
```

You will see `Hot-to-go` messages indicating devices on the PCU have been restarted and are ready to use.

12. (Optional) Back up site files using the `sitedump` command.

## Adding a Wire

To add a wire service to your system, you must complete four phases in the setup.

- Phase 1 - Connect the wire service to a PCU port.

- Phase 2 - Create an entry for the wire service in the configuration file and a wire profile for the wire service.
- Phase 3 - Add the wire distribution information.
- Phase 4 - Restart the wire service's program to incorporate the changes you made, so it can begin receiving and distributing wire stories.

You can edit just a wire service's profile.

For more information about wires, including the step-by-step procedure outlining four phases for adding a wire service, see "Adding a Wire to Your Avstar System" on page 13-2.

## Alternative Editing of the Site Configuration File

The UNIX line editor is used at the console to edit site files, which are located on an area of the server's hard disk known as the software partition. If these files, such as the `/site/config` file, are temporarily transferred to the Avstar database located on an area known as the database partition, you can use an Avstar Workstation to edit the file. Here is the alternative method of editing a site file, such as the `/site/config` file, in the database rather than using the UNIX line editor:

1. At an Avstar Workstation, create a transfer queue—that is, go to a directory, such as the System directory, and create a new queue to hold the configuration file. A `SYSTEM.TRANSFER` queue may already be set up on your system. See "Adding a Directory or Queue" on page 5-3 for more information.



*When performing this procedure, there should be only one file—the one you are working on—in `SYSTEM.TRANSFER`.*

2. At the Avstar console, use the `doc` command to transfer material between the software and database partitions.

The `doc` command format is:

```
doc -pu <queue name> <file to be transferred>
```

For instance, if you wanted to transfer a copy of the configuration file to the `SYSTEM.TRANSFER` queue, you would type:

```
doc -pu system.transfer /site/config
```

3. Return to an Avstar Workstation.
4. Log in as a system administrator. (This is to ensure that you have access to the System directory.)
5. Navigate to the `SYSTEM.TRANSFER` queue and open it by double clicking on it.
6. Select the configuration file story in the queue and edit it in the Story panel.



*If you are adding devices in the bottom section of the configuration file, do not forget to also add the device numbers in the hosts section at the top.*

7. Move file from database partition back to software partition using the `doc` command again. The format is:

```
doc -gu <queuename> > <filename>
```

For instance, to move the new configuration file from the `SYSTEM.TRANSFER` queue back into the Site directory, do the following:

```
doc -gu system.transfer > /site/config.new
```



*The example shows the file is transferred into a temporary location (`/site/config.new` rather than `/site/config`). This allows for testing prior to actual file implementation.*

8. Use the `configure` command to test the new configuration file, to see if there are any problems with it. If none are found, you will get the system prompt; otherwise, you will get an error message.

Here are some sample configuration tests:

```
configure /site/config.new abc a
```

```
configure /site/config.new ab a
```

```
configure /site/config.new a a
```

```
configure /site/config.new bc b
```

```
configure /site/config.new ac c
```

9. Copy the new configuration file into the correct location:

```
cp /site/config.new /site/config
```

10. Remote copy it to the other server(s):

```
rcp /site/config AVSTAR-B:/site/config
```

```
rcp /site/config AVSTAR-C:/site/config
```

11. Implement the new configuration by reconfiguring the system from the master computer (typically server A). To do this, do the following:

- a. Select the master computer.

- b. Take the system offline by typing:

```
AVSTAR-A: offline
```

- c. Reconfigure the system by typing:

```
AVSTAR-A: configure
```

- d. Take the system back online by typing:

```
AVSTAR-A: online
```

12. Return to the Avstar Workstation and delete the configuration file from `SYSTEM.TRANSFER`.



Usually, iNews Customer Support technicians edit the configuration file on the console, in the software partition. If you leave a copy of the file in the database, it is possible someone may change the file on the console, so when you come back to edit the file in the database (which you assume to be the more current version), it will actually be an outdated version. To eliminate the possibility of confusion, delete the file from the database when you are done.

## Intersystem Messaging

Intersystem Messaging is a feature of Avstar NRCS v.1.3 that allows a user to exchange messages with another user on a separate Avstar Newsroom Computer System, or other third-party system with a compatible interface.

On Avstar Newsroom Computer Systems, intersystem messages can be sent from Video Terminal (VT) or Avstar Workstation (ASWS) sessions, and from the Avstar console `send` utility. For intersystem messaging to work, a system must have an agent that functions as described in the following section. For Avstar NRCS, this agent is integrated into the Avstar Server software.

RFC (Request For Comments) documentation is provided at the following Web sites:  
<http://sunsite.auc.dk/RFC> or <http://www.rfc-editor.org/>

To receive intersystem messages, a system must have a TCPMUX service running. TCPMUX is defined by RFC 1078, "TCP Port Service Multiplexer (TCPMUX)." Additionally, the system must have an intersystem message service configured. For UNIX systems, the `/etc/inetd.conf` directory must contain these service definitions.

## Sending Intersystem Messages

An intersystem send is attempted whenever a message send request has a recipient name which includes an at symbol (@). It is assumed that this represents a name in `<user name>@<system name>` format. This is the same format used for sending mail to a user on a foreign system, such as the Internet. The `<system name>` parameter can be an IP address in standard notation, such as `172.161.131.2`.

The system name is resolved to an IP address through standard lookup services.

A TCP connection to port 1 of the system is attempted. Port 1 is the "well known port" (as defined in RFC 1700, "Assigned Numbers") assigned to the TCPMUX service.

Once the connection is established, the string `inter_system_message<cr-lf>` is sent. The receiving system sends `+<explanation><cr-lf>` to indicate a positive acknowledgement. This conforms to RFC 1078.



*The service name—in this case, `inter_system_message`—is never case-sensitive and `<explanation>` is any text that helps to explain the reason for the response.*

The sending system can then send the following string:

```
SEND<sp><user name><sp><sender's name><sp><message
text><cr-lf>
```

In this string, the user name and sender's name do not contain any spaces, carriage returns, or line feeds. The names are as they are used within their respective systems. The sender's name should be suitable to use as the user name in an intersystem message reply. The message text can contain spaces but not carriage returns or line feeds. It is optional (so you can check the logged-in status of the user). Avstar NRCS will truncate this string at the first line feed or at 72 characters.

After sending the intersystem message, the sending system should read a single line, which is the receiving system's response. On receipt of the response, the sending system should close its connection. See the description of responses in "Receiving Intersystem Messages" on page 11-53.

The sending system should be prepared to handle all of these error conditions:

- Time out on establishing connection to the receiving system's TCPMUX port
- The receiving system actively refusing the connection on the TCPMUX port
- Connection closed by the receiving system at any time
- A negative response `-<explanation><cr-lf>` to the `inter_system_message<cr-lf>` request. This is a negative

TCPMUX response, and `<explanation>` is any text that helps to explain the reason for the response.

## Receiving Intersystem Messages

To receive intersystem messages, a system must respond to connections on the TCPMUX port. On UNIX systems, this is done by having a TCPMUX service defined in `/etc/services` and `/etc/inetd.conf`.

To hook an intersystem message service into the TCPMUX service on a UNIX system, an entry must be included in the `/etc/inetd.conf` such as:

```
tcpmux/+inter_system_message stream tcp nowait
<agent> <parameters>
```

The `inter_system_message` string is the identifier used by the sending system to select this service. This string is not case sensitive.

The plus character (+) preceding `inter_system_message` tells the UNIX `inetd` daemon to handle the initial connection and negotiation. In this case, when the `inetd` daemon determines that it has an intersystem message agent, it will perform the positive acknowledgment (the `+. . . <cr-lf>` response) and then invoke the agent program.

The agent program must be prepared to respond to the “SEND” command, as described above.

The Avstar NRCS agent program is `/exc/ismessage`. The suggested parameter for the Avstar NRCS intersystem message agent is `ism`. This parameter can be anything and is used to identify the program in messages printed to the system’s console. If an additional parameter is supplied and it is a non-zero decimal string, the `/exc/ismessage` program is put into a verbose mode. When in verbose mode, the `/exc/ismessage` program will print its responses onto the system

console. This can be used to track frequency and identity of intersystem messages.



*The actual message text is not printed onto the console.*

So, for Avstar NRCS, the service line in `/etc/inetd.conf` for normal operation is:

```
tcpmux/+inter_system_message stream tcp nowait /exc/ismessage ism
```

Responses from the receiving agent program must conform to the following syntax:

```
<3-digit response code><sp><explanation><cr-lf>
```

The 3-digit response code is modeled on FTP response codes (See Section 4.2 of RFC 959, “File Transfer Protocol”).



*However, only single line responses are expected. The explanation is any text excluding a carriage return – linefeed (`cr-lf`), which makes the response better understood.*

The receiving agent program will generate one of the following responses, with the Avstar NRCS receiving agent including the following explanations:

```
201 <user name> is logged in<cr-lf>
```

Message stored for the specified user (if there is message text) and the user is currently logged in. User is notified of message arrival.

```
202 <user name> is not logged in<cr-lf>
```

Message stored for the specified user (if there is message text) and the user is not currently logged in.



421 System not online<cr-lf>

System not connected, not configured, or not online. Message is discarded.

430 No such user: <user name><cr-lf>

Username unknown on receiving system. Message is discarded.

450 Message save failed for: <user name><cr-lf>

Failed to properly store the message for the specified user. Message is discarded.

500 Syntax error, command unrecognized<cr-lf>

The first “word” on the line was not “send.” The check for the word “send” is case insensitive. Message is discarded.

501 Syntax error, insufficient parameters<cr-lf>

The “send” line has fewer than three space-delimited tokens. Minimally “send”, <user name>, and <sender’s name> are required, <message text> is optional. Message is discarded.

The Avstar NRCS receiving agent will print diagnostics to the system’s console when abnormal conditions are encountered. The diagnostics are:

ism: getpeername failed (<errno>) <errno string>

The sender’s IP address could not be determined.

ism: fgets error (<errno>) <errno string>

The read failed for the send command.

ism: gethostbyaddr failed (<errno>) <errno string>

The sender’s IP address could not be converted into a host name.

`Errno` is the UNIX system error number returned on system function calls and `errno string` is an explanation of that error code.

The Avstar NRCS receiving agent will accept intersystem messages directed to the user’s computer. Messages addressed to “computer” will be printed on the system’s console. The word “computer” can be localized using the message dictionary token `M_COMPUTER`. A 201

computer is logged in response will always be returned for messages directed to “computer.”

## Database Change

The Avstar NRCS v.1.3 message file format includes the sending system’s IP address for intersystem messages. When upgrading an Avstar Newsroom Computer System, a `dbdump` of the message file should be done to preserve the message file content. After upgrading, a `dbgen x` (`x` identifies the message file) console command must be done. A `dbrestore` can then be done to restore the message file.

## Avstar Workstation Session Behavior

There is virtually no difference between sending local messages and sending intersystem messages. If the recipient’s name contains an at symbol (@), name validation is not performed and an intersystem send is attempted. Possible responses are identical to those of Video Terminal (VT) sessions.

The only difference on receiving messages is that the complete sender’s information is always returned. If it is a local message, a simple user name is provided. If it is an intersystem message, the sender’s name will be formatted as `<sender’s name>@<system name>`.

## VT Session Behavior

The only difference between sending a local message and sending an intersystem message is the recipient name. If it contains an at symbol (@), an intersystem message send is attempted. Possible responses to an intersystem message send are:

<code>D_OFFLINE</code>	Received a 421 response.
<code>D_NOUSER</code>	Received a 430 response.
<code>D_NOLOG</code>	Received a 202 response.
<code>D_LOGGEDIN</code>	Received a 201 response.

D_BADDEST	Unknown host, or could not connect to host, or host refused connection, or no intersystem message service available on receiving system.
D_ERROR	A system error occurred.

There are minor differences when receiving intersystem messages. Local messages always have the sender's name enclosed in square brackets ([ ]). Intersystem messages substitute an at symbol (@) for the close bracket (]). That is the only visible difference when viewing the message on the command line.

Complete sender information, including the sender's name and system is included when messages are recalled (using the `message recall` command).



*Since the sending system's IP address is stored in the message, a hostname lookup determines the sending system's name. If the hostname lookup fails, the IP address is used.*

The VT has a reply function. It is invoked when the user enters the `send` command without any parameters. The sender's name of the last message retrieved is put onto the command line. The full user name and system name will be used. The system limits the length of the name to 41 characters. If the sender's name exceeds that length, the IP address (in standard notation) is used for the system name.



*The system stores IP addresses and will look up the system names when necessary.*



# CHAPTER 12

## *Printers*

Each printer on your system has a profile that contains the commands Avstar NRCS needs to control the printer, plus settings for options you may use with the printer. When you restart a printer, your system checks its profile for the necessary information. There are two types of printing that require configuration: system and local printing. Managing both are covered in this chapter. For information on how to connect a printer to Avstar NRCS, see “Adding a Printer” on page 11-41.

This chapter has the following sections:

- System Printing
  - The Printer Profile Files (in `/site/printers`)
  - Customizing Print Effects (Fonts)
  - Defining Print Forms
  - Printer Profile Options
  - Using Special Characters in a Profile
- Creating and Using Print Styles
- Managing Printers
- Local Printing
  - Local Printing Dialog Box
  - Local Print Style Options

## System Printing

A *system printer* is connected to an Avstar Server and not directly connected to an Avstar Workstation. Users can send print jobs to a system printer from any Avstar Workstation on the network. System printers can be customized by configuring certain system profile files and forms.

### The Printer Profile Files (in /site/printers)

Printer profiles are text files in the `/site/printers` directory. Your system is installed with standard profiles for many popular printers. To see which printer profiles are installed on your system, go to the Avstar console and type:

```
ls /site/printers
```

Information similar to the following appears:

15	81	facit	laser2	rawprint
17	95	generic	laser3	ti820
25	adx	genicom	laser4	ti850
35	apcarbon	hplaserjet	magnum2rev	ti855
45	apcarbonra	la120	nec	ti880
55	autocue	laser	okidata	today
65	epson	laser.bold	printronix	today-script

This list contains numbered and alphanumeric filenames. Alphanumeric files are the standard profiles for several different types of printers. For instance, `ti855` is the standard printer profile for the TI-855 printer. Numbered files are profiles your system uses to operate your printers.

Numbered files in the list correspond to device numbers of printers in your system. For the system to find the correct profile for each printer, the profile must be stored in a numbered file corresponding to the printer's device number. For instance, a printer whose device number is 17 must have its profile in a file named `/site/printers/17`.

The standard profile for an HP LaserJet printer is in `/site/printers/hplaser`. To list this file, type:

```
cat /site/printers/hplaserjet
```

Information similar to the following appears:

```
AVSTAR-A: cat /site/printers/hplaserjet
;HP Laserjet IVsi 14SEP00
;
ejectcode      <ff>
ejectcount     1
idlecount      0
pagelength     66
scriptrhstart  32
scriptrhmax    23
scriptlhmax    25
scriptshift    yes
scripttemplate no
;
expand         <esc>(0U<esc>(sp7.0h18vs0b11T   <nul>   ;Courier
;
font 2         <esc>(s3B           <esc>(s0B           ;Bold
font 3         <esc>(s4B           <esc>(s0B           ;extra bold
font 4         <esc>&dD           <esc>&d@           ;underline
font 5         <esc>&dD<esc>(s3B   <esc>&d@<esc>(s0B   ;bold/Under
;
form 1         <esc>E             <nul>             ;reset
form 7         <esc>&l00<esc>(s10H<esc>(s12V   <nul>       ;landscape
;form 8         <esc>&l1H           <nul>             ;top tray
;form 9         <esc>&l14H         <nul>             ;lower tray
;
```

The first half of the file contains options that control margins, headers, form feeds, and page length. The second half contains fonts and forms. The fonts section defines control codes that select print effects, such as bold or underlining. The forms section defines codes that set up the printer, such as control codes to select a font module or set print quality.

Control codes often use nonprinting characters such as the one generated by the ESC key. All control codes in the profile above begin with the character produced by the ESC key. Because this key does not produce a printing character, enter it in the file as `<esc>`.

## Customizing Print Effects (Fonts)

To turn print effects—such as bold or underlining—on and off, the Avstar system must send control codes to the printer. Control codes are defined as `font` in a printer profile.

A profile can have up to 10 fonts, numbered from 1 to 10. Each font defines the control code to turn on an effect and the control code to turn off the effect. Usually, a font defines the control codes for one effect. However, you can combine control codes for effects, such as bold and underlining, to create complex fonts.

### Defining a Font

Each font is defined on a separate line in a printer profile. The definition must begin with the word `font` followed by the font number, which must be between 1 and 10. This is followed by the control code that turns on the effect, a few spaces, and the control code that turns off the effect.

Style stories select fonts by number, and any printer can use them. If you have a style story that expects `font 1` to represent bold text, define `font 1` as bold in all your printer profiles.

For instance, to create a bold font, start a line with `font 1`, followed by the code to turn on bold printing. For an HP laser printer, the code would be `<esc>(s3B`. Follow that with a few spaces and the code to turn off bold printing `<esc>(s0B`. The finished definition looks like this:

```
font 1    <esc>(s3B      <esc>(s0B      ;bold print
```



*Assign fonts consistently in all your printer profiles, such as font 1 is bold; font 2 is underlined, and so on.*

### Combining Print Effects

In many cases, you can combine two or more print effects by creating a font containing the print effects control codes. Follow the font name with control codes to turn on the print effects you want. Follow these



codes with a few spaces and control codes to turn off those print effects.

For instance, the HP LaserJet printer can print both bold and underlined text. It uses `<esc>(s3B` and `<esc> s0B` to turn bold on and off and `<esc>&dD` and `<esc>(&d@` to turn underline on and off.



*When you combine several print effects in one font, no spaces can occur between control codes that turn on all print effects. Additionally, no spaces can occur between control codes that turn them off.*

For instance, combine two effects to create a font, such as font 5, that prints bold and underline as follows:

```
font 5    <esc>&dD<esc>(s3B    <esc>&d@<esc>(s0B
;bold, underline
```

## Defining Print Forms

When you send a story to a printer, your system must first initialize the printer to prepare it to print according to the style you select. Define each of the control codes that represent initialization commands as `form`. For instance, define codes that select draft quality printing as `form 1` and codes that select letter quality printing as `form 2`. You can define up to 10 forms in a printer profile. See “Selecting Forms” on page 12-26 for more information.

Each form contains on and off control codes for a different setup command. Usually, a form contains control codes for one setup command, but you can also combine control codes for different commands, such as select both letter quality and a font module.



*Forms may seem to be the same as fonts, but forms are used strictly to initialize the printer. Therefore, use a form to set a printer to word processing mode before printing, but not to turn on bold printing.*

## Defining a Form

You define forms similarly to fonts. Each definition begins with the word, “form,” followed by the form’s name. The form’s name contains a number between 1 and 10. This is followed by a control code that initializes the printer, a few spaces and, optionally, an `off` control code.

The form’s `on` control code is sent at the beginning of the print job, and the form’s `off` control code is sent at the end of the job.

Each setup option has an `on` control code. Many—but not all—have an `off` control code. If the setup option you are defining does not use an `off` control code, you must put a `<nul>` where the `off` control code would be located.

Always name forms consistently in each printer profile. For instance, in each profile define `form 1` as draft quality.

Suppose that you are creating a profile for your HP laser printer and want to define a form that sets up the printer in landscape mode. Define this as `form 7`.

Begin the line with `form 7`, and follow with the control code that puts the printer in landscape mode. For an HP laser printer, the control code is:

```
<esc>&10o<esc>(s10H<esc>(s12V
```

The HP laser printer needs landscape mode turned off at the end of a print request, so follow the `on` control code with `<esc>E`. The finished line should look like this:

```
form 7    <esc>&10o<esc>(s10H<esc>(s12V    <esc>E  
;landscape
```

## Combining Setup Options

As with fonts, you can combine two or more setup options in one form. You do this in the same way you would combine options for fonts. See “Combining Print Effects” on page 12-4 for more information.

## Font and Form Space Available

The system reserves about 400 bytes of space for form and font definitions for each system printer profile. Each character in a font or form definition—including the spaces to separate `on` and `off` control codes and the Enter character at the end of each line—uses space.

If you exceed the limit, the message, `Maximum Definitions Already Made`, appears when you try to start the printer.

If you see this message, free some space by deleting forms or fonts from that profile that you do not use. Alternatively, comment out any forms and fonts you are not using by placing semicolons in front of them.

## Printer Profile Options

Besides defining forms and fonts, you can set a number of options in the profile to be used when a print request is sent to the system printer. These options control formatting, such as the number of lines on each page, whether or not the text is all uppercase, whether stories have a banner at the top of each page, and so on.

These options fall into two categories:

- Profile-only options are set in the printer profile.
- Profile and style options are also set in the printer profile, but can be temporarily overridden if a different value is specified in a style story within the database.

Almost all of these options have default values the system uses if you do not set the options.

For options pertaining to local printing, see “Local Print Style Options” on page 12-38.

## Profile-Only Options

Profile-only options are set in the printer profile, and cannot be overridden in a style story. Use options beginning with the word `auto` only with an autocue printer, which produces teleprompter roll copy according to European standards—that is, printing only the audio portion of a script. The underlined item or number in parenthesis is the default value.

The profile-only options are described in Table 12-1.

**Table 12-1 Profile-Only Options**

Option	Description
<code>answerback &lt;message to send&gt; &lt;expected reply&gt;</code>	Use with printers that require the system to send a specific message and then expect a certain answer back from the printer before sending the story to the printer. There is no limit on these strings.
<code>autofield &lt;form field&gt; (S)</code>	Determines which form field of the story is used as a header line when printing the story. The default value is the title field (S). This is a Video Terminal (VT) template field identifier character, not a field name.
<code>autoindent &lt;number of spaces&gt; (0)</code>	Sets the number of spaces to indent each line (as a printing offset) on an autocue printer. The default value is 0.
<code>autolength &lt;length of line&gt; (30)</code>	Sets the maximum length for lines of text on an autocue printer. The default value is 30.
<code>autopara &lt;number of spaces&gt; (3)</code>	Controls the number of spaces an autocue printer indents each paragraph. The default value is 3 spaces.
<code>autospace &lt;line spacing&gt; (3)</code>	Controls line spacing for an autocue printer. A value of 1 produces single-spaced text. A value of 2 produces double-spaced text. A value of 3 is the default and produces lines of text separated by two blank lines.



**Table 12-1 Profile-Only Options (Continued)**

Option	Description
autoshift [ <u>y</u> es   no]	Controls whether an autocue printer prints in uppercase or mixed case. When the value is yes, it uses uppercase. If set to no, text is printed as entered. The default value is yes.
done <queue name> ( <u>D</u> ead)	Specifies the queue where a print request is sent after the print job is done. The default value is the Dead queue.
ejectcode <printer's eject string>	Lets the system know which codes it must send to the printer to execute a form feed. There is no default for this option, so you must explicitly define the eject code in the printer profile.
ejectcount <number of ejects> ( <u>1</u> )	Selects the number of times your system sends the eject code to the printer at the end of each print request. The default is 1.
exlines <number of lines> ( <u>0</u> )	Printers that print double-high text, such as the Print-ronix P300, use one extra line per line of expanded text. Use this option to set the number of additional lines used when printing expanded text with the <code>print script</code> command. The default is 0.


**Table 12-1 Profile-Only Options (Continued)**

Option	Description
expand <start string> <end string>	<p>Defines control codes that turn a printer's expanded text mode on and off. The system uses this option when someone prints a story using the <code>print script</code> command. If you do not define these control codes in a printer's profile, or if a printer cannot print expanded text, the system prints scripts in normal text.</p> <p>To set up, begin a line with <code>expand</code>. Follow that with your printer's control codes to turn on expanded printing, a few spaces (or a tab space), and the control codes to turn off expanded printing. For instance, if the code to turn on expanded printing is <code>&lt;esc&gt;P</code> and the code to turn it off is <code>&lt;esc&gt;Q</code>, define expanded printing like this:</p> <pre>expand &lt;esc&gt;P &lt;esc&gt;Q ; (DP) expanded</pre>
font	<p>Defines control codes that turn on or off print effects, such as bold or underlining. See "Customizing Print Effects (Fonts)" on page 12-4 for more information.</p>
form	<p>Defines control codes that represent initialization commands, such as selecting draft quality. See "Defining Print Forms" on page 12-5 and "Selecting Forms" on page 12-26 for more information.</p>
idlecount <number of ejects> (0)	<p>Determines the number of ejects (as defined by the <code>ejectcode</code> and <code>ejectcount</code> options) to be performed once all pending print requests have been printed. The default value of 0 prevents the system from ejecting a page at the end of a print request.</p>
initprint <printer's initialization string>	<p>The system uses these codes to initialize the printer when you restart it. This may be lost when the printer is powered off.</p>

**Table 12-1 Profile-Only Options (Continued)**

Option	Description
map <database character> <printer code>	<p>Use to translate or map a database character to a character the printer can print. Useful if you need characters such as <math>\grave{\text{i}}</math> and <math>\text{£}</math>, which are available on many printers but have a different code in the printer.</p> <p>To map a translation of a database character, begin a line with <code>map</code>. Add a few spaces and enter in the database character you want to translate. Add a few more spaces and enter the code you want the system to send to the printer in place of that database character.</p> <p> <i>The printer code is a single byte and not a sequence of bytes.</i></p>
sbc_frames <frames per second>	<p>Indicates the number of video frames transmitted in one second. This value is used when calculating the duration of a tape event. There is no restriction on this value. However, Phase Alternate Lines (PAL) video format transmits 25 frames per second and National Television Systems Committee (NTSC) video format transmits 30 frames per second. This Sony barcode parameter is required.</p> <p> <i>If printer type is Sony-beta, sbc-frames must be defined. Otherwise, a Missing required barcode parameter message will appear.</i></p>

**Table 12-1 Profile-Only Options (Continued)**

Option	Description
sbc_max_duration <[hh:]mm:ss:ff>	<p>This Sony barcode parameter has two functions. It is used to determine the maximum segment duration time if raw time code is being used to calculate the segment duration, which is calculated as the End Of Message (eom) time code minus the Start Of Message (som) time code. It is also used to check if a duration time entered from the template does not exceed the duration entered in the profile. In the format, <code>hh</code> is hours, <code>mm</code> is minutes, <code>ss</code> is seconds, and <code>ff</code> is frames. The maximum time allowed is <code>23:59:59:ff</code>, where <code>ff</code> is the value found in <code>sbc_frames</code>. This Sony barcode parameter is required.</p> <p> <i>If printer type is Sony-beta, sbc_max_duration must be defined. Otherwise, a Missing required barcode parameter message will appear.</i></p>
sbc_som <hh:mm:ss:ff>	<p>Enables the user to determine a constant Start Of Message (som) timecode. The maximum value permitted is <code>23:59:59:ff</code>, where <code>ff</code> is the value found in <code>sbc_frames</code>. This Sony barcode parameter is optional.</p>
sbc_eomstyle [<HH:mm:ss:ff>   <hh:mm:ss:FF>]	<p>Enables a user to specify a constant End Of Message (eom) style. A user can declare a constant hour field or a constant frame field. It is not possible to declare both. The format ends with either <code>HH:mm:ss:ff</code>, where <code>HH</code> is the constant hour field value or <code>hh:mm:ss:FF</code>, where <code>FF</code> is the constant frame field. The <code>mm</code> and <code>ss</code> parts of the format are literal. If you specify a number of hours, you must put <code>ff</code> for frames. If you specify a number of frames, you must put <code>HH</code> for hours. This Sony barcode parameter is optional.</p>
sbc_bulkprint	<p>Permits printing of multiple Sony barcode labels, when the command <code>print story</code> is entered from the directory level.</p>



**Table 12-1 Profile-Only Options (Continued)**

Option	Description
scriptshift [ <u>yes</u>   no]	Controls whether scripts are printed in uppercase. The yes option prints scripts in uppercase; no prints scripts as is. The default is yes.
type [ <u>generic</u>   autocue   adx   sony_beta   beeline]	Defines European Autocue, ADX, generic, beeline, and Sony Barcode printers. If the profile is for any of these printers, use this option to specify the printer. The default value is generic.

## Profile and Style Options

The options in Table 12-2 are set in the printer profile, but can be overridden by settings in a style story. Generally, use a printer profile to set these options to the values you want the system to use as defaults for a particular system printer. When you create a style story, you need only include these options to use a value different from the one defined in the printer profile.

**Table 12-2 Profile and Style Options**

Option	Description
banner [ <u>yes</u>   no   continuous]	<p>Controls whether each page begins with a header that includes the name of the user who printed the story, the time, and the page number. If you want a banner, set this option to yes. The default value is yes.</p> <p>When you print a queue, your system numbers each story in the queue, beginning with page one by default. To make the page numbers continuous, set the banner option to continuous.</p>
blanks <number of blank lines> (3)	Controls the number of blank lines the system prints following each story when printing a directory. A value of 1 results in single spacing (no blanks between stories), a value of 2 gives double spacing, and so on. The minimum value is 1. The default value is 3.

**Table 12-2 Profile and Style Options (Continued)**

Option	Description
pagefooter <queue name> [<footer name>]	Use to include a footer on each page of printed output.
pageheader <queue name> [<header name>]	Use to include a header on each page of printed output.
pagelength <number of lines> (65)	Determines the number of lines printed on a page before the page is ejected. Specify a value of 1 for no page breaks. Other valid values are 6 through 255. The system uses the default if you specify an invalid value. The default value is 65.
printform [ <u>yes</u>   no   all]	Controls whether a story's form is included in the printed output. The <code>yes</code> option prints the form at the top of the first page only; <code>no</code> prints the story without a form; and <code>all</code> prints the form at the top of each page of the story. The default value is <code>yes</code> .
printscriptform [ <code>yes</code>   <u>no</u>   all]	By default, a script is printed with only the title, page, and presenter fields at the top of the first page. To print the entire form on the first page, set this option to <code>yes</code> . The <code>printscriptform all</code> option causes the entire form to be printed on each page of the script. The <code>printscriptform no</code> option causes the script to be printed without any form information.
scriptlmax <number of columns> (28)	Sets margin of left (production cue) portion of a script. Adjust this value if you are changing from a script printed in normal text to a script printed in expanded text. The default value is 28.
scriptrhmax <number of columns> (25)	Controls the right margin of a script. Normal settings are from 20 to 30. Set this option to the same value as the <code>airmargin</code> parameter in the system profile to ensure that the <code>print script</code> command prints a script that looks the same as the script displayed at the workstation with the <code>air</code> command. The default value is 25.

**Table 12-2 Profile and Style Options (Continued)**

Option	Description
scriptrhstart <column number> (29)	Sets starting column of right (audio) portion of a script. If you are using normal-sized print, column 40 is the midpoint of a page and the value generally used. If you use expanded print, use a smaller value to accommodate the larger text. The <code>scriptrhstart</code> printer-profile option differs from the <code>scriptrhstart</code> parameter in the system profile. The default value is 29.
scriptspacing <number of lines> (2)	Sets spacing used between lines of printed text when printing a script. Use a value of 2 for double-spacing. Use a value of 1 (the minimum value) for single-spacing. The default value is 2.
separator [ <u>yes</u>   no]	Specify whether a row of equal (=) characters is printed after a story's form to separate it from the text of the story. The default value is yes.
spacing <line spacing value> (2)	Controls line spacing of printed stories. Set the value to 1 for single-spacing, or 2 for double spacing. The default value is 2.

## Standard Header and Footer Options

Use the `pageheader` and `pagefooter` options to include headers and footers in printed stories. You can put your header and footer stories anywhere in your database. Group header stories in a queue called `SYSTEM.HEADERS` and group footer stories in a queue called `SYSTEM.FOOTERS`.

To create a header or footer, enter the text into the story as you want it to appear on the printed page. The total number of lines in the header and footer combined cannot be more than half of the `pagelength` value.

For instance, if the `pagelength` is 66, the combined number of lines in the header and footer cannot be more than 33. If the header/footer

combination exceeds half of the `pagelength` value, the system truncates header and footer lines so that both fit on the page.

To select a header in a printer profile or a style story, use the `pageheader` option. The format for this option is:

```
pageheader <queue name> <header name>
```

The queue name is the full pathname of the queue containing the header story you want to use. The header name is generally the title of the story in that queue containing the header. It is defined by the Header queue's `sorting index` field, whose default is the `Title` field. For instance, to select a header from `SYSTEM.HEADERS` called *am-script*, use the following:

```
pageheader    system.headers    am-script
```

To select a footer story in a printer profile or a style story, use the `pagefooter` option. The format for this option is:

```
pagefooter <queue name> <footer name>
```

The footer name is generally the title of the story in that queue containing the footer. It is defined by the Footer queue's `sorting index` field, whose default is the `Title` field. For instance, to select a footer from `SYSTEM.FOOTERS` called *am-script*, use the following:

```
pagefooter    system.footers    am-script
```

Profiles and styles interpret spaces as delimiters; give your header and footer stories titles that do not include spaces and left-justify them in the title field.

## User-Selected Headers and Footers

You can also let the user select the header or footer when the story is printed. To enable this, the story must have a form field where the user can enter the header or footer name. You can use this method only with the `PRINT STORY` and `PRINT SCRIPT` commands.

To set up a profile or a style so the user can select a header or footer, use the `pageheader` or `pagefooter` options. Put an equal sign followed by a field name as the header or footer name.

For instance, suppose your Scripts queue has a form with a field called *footer* where users can enter the title of the footer story they want to use. Set up the `pagefooter` option in the printer's printer profile:

```
pagefooter      system.footers      =footer
```

When someone uses that printer to print a story, the system looks for a story in `SYSTEM.FOOTERS` with the name entered in the story's footer form field. The name must be an actual story in the queue named in the `pagefooter` option, or the footer is not printed.

## Profile Option Defaults

Many profile options have default values. These options and their default values are shown in Table 12-3. Some of these profile options also have minimum values. Options for which defaults or minimum values do not apply are marked n/a.

**Table 12-3 Profile Option Defaults**

Option Name	Default	Minimum	Maximum
answerback	n/a	n/a	n/a
autofield	S	n/a	n/a
autoindent	0	0	n/a
autolength	30	1	n/a
autopara	3	0	n/a
autoshift	yes	n/a	n/a
autospace	3	1	n/a
banner	yes	n/a	n/a
blanks	3	1	n/a
done	n/a	n/a	n/a
ejectcode	n/a	n/a	n/a

**Table 12-3 Profile Option Defaults (Continued)**

<b>Option Name</b>	<b>Default</b>	<b>Minimum</b>	<b>Maximum</b>
ejectcount	1	0	n/a
exlines	0	0	n/a
expand	n/a	n/a	n/a
idlecount	0	0	n/a
initprint	n/a	n/a	n/a
map	n/a	n/a	n/a
pagefooter	n/a	n/a	n/a
pageheader	n/a	n/a	n/a
pagelength	65	1	255
printform	yes	n/a	n/a
printscriptform	no	n/a	n/a
scriptlhmax	28	1	79
scriptrhmax	25	1	79
scriptrhstart	29	2	79
scriptshift	yes	n/a	n/a
scriptspacing	2	1	n/a
separator	yes	n/a	n/a
spacing	1	1	n/a

**Table 12-3 Profile Option Defaults (Continued)**

Option Name	Default	Minimum	Maximum
type	generic	Setting this parameter to autocue/adx/sony_beta/beeline changes the program behavior for a special purpose.	

## Using Special Characters in a Profile

Some printers use control codes that include nonprinting characters, such as those produced by the Escape and Enter keys.

Others use characters such as semicolons (;), pound signs (#), and at (@) symbols, which have a special meaning for your system or your system's console.

You must enter these characters in your profile in a special way.

### Using Nonprinting Characters

A nonprinting character is one that does not show up on your screen. You can add a nonprinting character to a profile either by alias or by decimal value.

#### Adding Nonprinting Characters by Alias

Your system recognizes a number of aliases for many of the nonprinting characters you may use in a printer profile. These aliases are two- or three-letter abbreviations that represent different characters. For instance, `esc` represents Escape (ESC) and `cr` represents Enter.

When you put an alias in a printer profile, you must enclose it in angle brackets (< >) so that your system recognizes it as an alias. For instance, enter a font that used the Escape and Enter characters as follows:

```
font 3 <esc>4 <cr>5
```

### Adding Nonprinting Characters by Decimal Value

You can represent any character by its decimal value. Enter the character's decimal value in angle brackets (< >) where you want it to appear in the profile.

For instance, Escape has a decimal value of 27, and Enter has a decimal value of 13. Accordingly, you could enter a font using these characters in its control codes as follows:

```
font 3    <27>4    <13>5
```

See Table 12-5, "Character Alias and Decimal Values," on page 12-21 for more information.

### Avoiding Characters Used by the System

Some printing characters are reserved for system use. Therefore, you cannot include any of these characters as a control code by typing it. Some of these characters and their system interpretations are listed in Table 12-4.

**Table 12-4** Characters Used by the System

Character	How it is interpreted
;	Indicates the start of a comment
#	Represents the computer's name
space	Separates on and off control codes
<	Indicates the start of character entered by alias or decimal value
@	Causes system to abandon the line you are typing

If your printer uses control codes that contain any of these characters, enter the characters by alias or by decimal value. The aliases and decimal values for these characters are listed in Table 12-5.



**Table 12-5 Character Alias and Decimal Values**

Character	Alias	Decimal Value
;	<sc>	<59>
#		<35>
space	<sp>	<32>
<	<<	<60>
@	\@	<64>
	<vrl>	<124>
\	<bks>	<92>
<ctrl>A	<soh>	<1>
<ctrl>B	<stx>	<2>
<ctrl>C	<etx>	<3>
<ctrl>D	<eot>	<4>
<ctrl>E	<eng>	<5>
<ctrl>F	<ack>	<6>
<ctrl>G	<bel>	<7>
<ctrl>H	<bs>	<8>
<ctrl>I	<ht>	<9>
<ctrl>J	<lf>	<10>
<ctrl>K	<rt>	<11>
<ctrl>L	<ff>	<12>

**Table 12-5 Character Alias and Decimal Values (Continued)**

<b>Character</b>	<b>Alias</b>	<b>Decimal Value</b>
<ctrl>M	<cr>	<13>
<ctrl>N	<so>	<14>
<ctrl>O	<si>	<15>
<ctrl>P	<dle>	<16>
<ctrl>Q	<dc1>	<17>
<ctrl>R	<dc2>	<18>
<ctrl>S	<dc3>	<19>
<ctrl>T	<dc4>	<20>
<ctrl>U	<nak>	<21>
<ctrl>V	<syn>	<22>
<ctrl>W	<eth>	<23>
<ctrl>X	<can>	<24>
<ctrl>Y	<em>	<25>
<ctrl>Z	<sub>	<26>
<ctrl>[	<esc>	<27>
<ctrl>\	<fs>	<28>
<ctrl>]	<gs>	<29>
<ctrl>^	<rs>	<30>
<ctrl>_	<us>	<31>

**Table 12-5 Character Alias and Decimal Values (Continued)**

Character	Alias	Decimal Value
<ind>	<ind>	<132>
<nel>	<nel>	<133>
<hts>	<hts>	<136>
<ri>	<ri>	<141>
<ss2>	<ss2>	<142>
<ss3>	<ss3>	<143>
<dcs>	<dcs>	<144>
<csi>	<csi>	<155>
<st>	<st>	<156>

## Creating and Using Print Styles

For your printer profiles to be useful, you must have styles designed to take advantage of them. Each of your system's styles is defined in a separate story in your database, called a *style story*. Each style story creates a particular kind of printed output by selecting forms and fonts necessary to produce the output. It may also override some options set in the printer's profile. In addition to using style stories to control printed output, you can change default translations of special system words that control print operations.

Print styles can be used to define printed output for both system and local printers. Local printing style stories are created in the same location and in the same manner as those for system printing. See "Local Printing" on page 12-31 and "Local Print Style Options" on page 12-38 for more information.

## Creating a Style Story

Each style story is held in its own queue in `SYSTEM.STYLES`. Queues in this directory have names that are three-digit numbers, such as 001. This lets you refer to each style by a number. For instance, a user can choose the print style defined in the style story in queue 001 by selecting that queue.

Each three-digit queue name in the example can be followed by a hyphen and a descriptive name, such as `SINGLE-SPACE`. This optional name is used by Avstar's Print dialog box.

To create a new style story and queue, do the following:

1. Create the new queue in `SYSTEM.STYLES`. See "Creating a New Queue" on page 5-6 for more information.
2. Choose the next available three-digit number for the queue name. For instance, if 004 is the last queue name used, create a style 005. Give the new queue a numerical name, such as `005-memo`. The number is followed by a hyphen and the descriptive word, `memo`.



*The largest number that you can use as a queue name is 255 and the lowest is 0. (Style 0 is the default style your system uses when someone enters a `print` command without specifying a style.) You can have as many as 250 separate style stories.*

3. Once the queue is created, open it by double-clicking on it in the Directory panel.
4. Create a new story to hold the style you want to create. See "Creating a New Story" on page 5-9 for more information.
5. Create the new style by writing the various profile options, font, and form information in the Story panel. This is explained in more detail in the next sections.
6. Save the story.

## Changing System Profile Options

The first part of your style story should contain options that are different from the default values set in the printer profile. For instance, suppose that profiles usually have the banner or running header option set to `yes`, the spacing option set to 2 (double spacing), and a page length of 65 lines. However, you want the memos to be single-spaced, printed without a banner, with a maximum of 40 lines per page.

Change these options in the style story. Put `banner no` on the first line of the style story, `spacing 1` on the second line, and `pagelength 40` on the third.

Setting these options in a style story overrides their settings in the printer profile, but only when this style is used. You can modify profile and style options only in the style story; you cannot modify profile-only options.

The style options are different for local printing. See “Local Print Style Options” on page 12-38 for more information.

These are the system profile options you can change in a style story for system printing:

```

banner [yes | no | continuous]
blanks <number of blank lines>
pagefooter <queue name> <footer name>
pageheader <queue name> <header name>
pagelength <number of lines>
printform [yes | no | all]
printscriptform [yes | no | all]
scriptlhmax <column number>
scriptrhmax <column number>
scriptrhstart <column number>
scriptspacing <number of lines>
separator [yes | no]
spacing <line spacing value>

```

## Selecting Forms

What appears next in the style story are forms selected to produce the output you want. For instance, to print memos in letter quality mode, choose forms that set up the printer to do that.

As part of this example, suppose that we must initialize letter quality and word processing modes on printers to produce letter quality output. Also, our system's printer profiles define codes to initialize letter quality printing and codes to initialize word processing mode as `form 1` and `form 2`, respectively.

To add forms to the style story, do the following:

1. Type `form` and the numbers representing forms you want to add. For instance, to add `form 1` and `form 2` to the style story, type `form 1 2`.
2. Add a comment at the end of each line explaining the form's purpose. Begin the comment with a semicolon and then enter a short explanation. If you use more than one form, ensure the forms do not conflict.



*The same word, `form`, is used in both the style story and printer profile, but have different, although related, meanings. A `form` in the printer profile defines a sequence of codes to be output to the printer when one of the 10 possible forms is selected. When specifying a `form` in a style story, you are selecting a list of forms to use that correspond to form numbers in the printer profile. See "Defining a Form" on page 12-6 for more information.*

## Identifying and Selecting Fonts

The fonts come last in the style story. This parameter allows you to match video effects produced by the Avstar Workstation (ASWS) to corresponding print effects. For instance, you can match the bold video attribute, which produces bold text on the screen, to the print effect that produces bold printed text.

This is a multilevel process.

Video attributes produced by ASWS are converted to video modes used for system printing. These video modes can be associated with font numbers, in the style story. Fonts, defined and associated with a number in the printer profile, produce the desired print effect on paper.

There are a total of five video modes available for system printing. Their standard names are:

- NORMAL
- BOLD
- REVERSE
- UNDERLINE
- BOLDREVERSE

Listed in Table 12-6 is a chart displaying some of the different video attributes that can be produced on ASWS and the associated video modes.

**Table 12-6 Video Attributes and Video Modes**

<b>Attribute</b>	<b>Video Mode</b>
NORMAL	NORMAL
BOLD	BOLD
ITALIC	REVERSE
UNDERLINE	UNDERLINE
PRESENTER	REVERSE
CLOSED CAPTION	BOLDREVERSE
BOLD ITALIC	BOLDREVERSE
BOLD UNDERLINE	BOLD

**Table 12-6 Video Attributes and Video Modes**

Attribute	Video Mode
ITALIC UNDERLINE	REVERSE
BOLD ITALIC UNDERLINE	BOLDREVERSE

Video modes are defined in a system dictionary file called `ccuvideo`, located in the `/site/dict/` directory. An example of a typical `ccuvideo` dictionary is shown here:

```
AVSTAR-A: cat /site/dict/ccuvideo
;
; %Z% %M%           %I%           Date %E%           %Q%
;
;                   NEWS Video Steps
;
;Standard Name     Usual translation
NORMAL             /
BOLD               /newscaster
REVERSE           /video
UNDERLINE         /underlined
BOLDREVERSE       /director
$$
```

The `ccuvideo` dictionary file is made up of two parts. On the left are standard names for five video modes. On the right, following a forward slash (/), are usual translations associated with those standard names.

As an example, we will map bold and underlined attributes on ASWS to corresponding print effects. The printer profiles in our example system are standardized so that they all define `font 1` as producing bold printed text, and `font 3` as producing underlined text. We assign bold to `font 1` and underline to `font 3`.



When you assign a video mode to a font in a style story, you must refer to the video mode by its standard name that your system recognizes for system printing. Usual translations are names that describe different video modes on a Video Terminal (VT) display screen.

To assign a video mode to a font in a style story, begin a line with the video mode's standard name followed by the font number you want to associate with that video mode. For instance, to associate the bold video mode with `font 1`, type a line in the style story similar to this:

```
bold 1 ; bold text
```

We assign the underline video mode to `font 3` in the same way, such as:

```
underline 3 ; underline text
```

When you save the story, the system incorporates the style story into its operations so users can select that style when they print stories from the database.

The bold video mode is also associated with the bold attribute on the Avstar Workstation (ASWS). The underline video mode is associated with the underline attribute on ASWS. When you type some bold text in a story in ASWS, that bold text is associated with the bold video mode. When you print that story using a style story, which associated bold video mode with `font 1`, the bold text you typed into your story on ASWS will print bold-faced on paper.

There is one more important system dictionary file, which determines how attributes of field labels and field text in Story forms are associated with the five standard video modes. This is called `convertvideo` and exists in the `/site/dict` directory.



*In some cases, the `/site/dict` directory may not be present on the system.*

The standard form for this file is shown here:

```

AVSTAR-A: cat /site/dict/convertvideo
;                attribute  mode
;
field-style-to-video  <b>                bold
field-style-to-video  <b><u>              bold
field-style-to-video  <b><i>              boldreverse
field-style-to-video  <b><i><u>          boldreverse
field-style-to-video  <i>                reverse
field-style-to-video  <i><u>            reverse
field-style-to-video  <u>              underline
label-style-to-video  <b>                bold
label-style-to-video  <b><u>              bold
label-style-to-video  <b><i>              boldreverse
label-style-to-video  <b><i><u>          boldreverse
label-style-to-video  <i>                reverse
label-style-to-video  <i><u>            reverse
label-style-to-video  <u>              underline

```



It is strongly recommended that you *not* change this file.

The letters in angle brackets (< >) represent attributes. For instance, <b> represents the bold attribute. The <i> represents italic. The <u> represents the underline attribute. Words in the right column represent video modes associated with attributes. This file affects how Story forms are displayed on video display terminals and how they are printed to system printers.

### Using Styles with Local Printing on Video Terminal Only

Some workstations have a printer connected to their auxiliary ports. To use this type of printer, include `on 0` in the command line when using any of the `PRINT` commands. This selects printer 0, which is the printer attached to the user's workstation. To create a command line that uses style 123 on a printer connected to an auxiliary port, type:

```
Print story style 123 on 0
```

You can specify a style when you print to printer 0, as long as a print server has been configured on your system.

## Local Printing

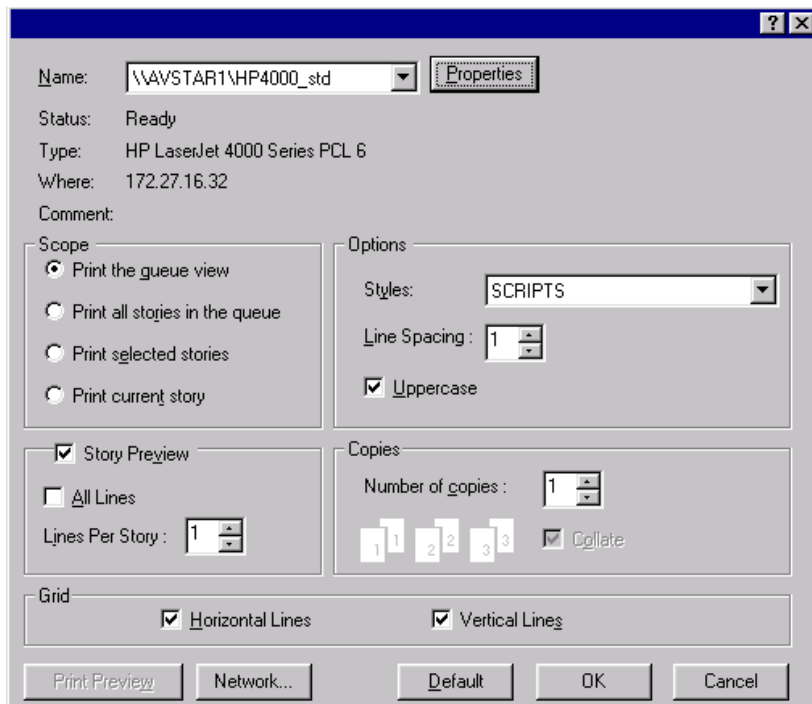
Local printing is any print request sent to a printer directly accessible to the Avstar Workstation.

Avstar NRCS includes an enhanced Local printing feature, also called Windows printing. These enhancements include a dialog box that offers users more options, such as selecting a predefined style when printing locally. Local printing does require some setup preparation or alterations. For instance, there are print style options that must be defined in `SYSTEM.STYLES`. Both print style options and the dialog box are explained here, starting with the dialog box.



*Local printing is not available when the cursor is in the Directory panel, unless it is on a queue that is currently displayed in the Queue panel. Local printing is not recommended for stories larger than 10 kilobytes or 5 minutes, according to Avstar timing. Avstar will print these files, but because the local printing is not handled in the background, it may cause the workstation to appear “frozen” or “locked up” while the system processes the print request.*

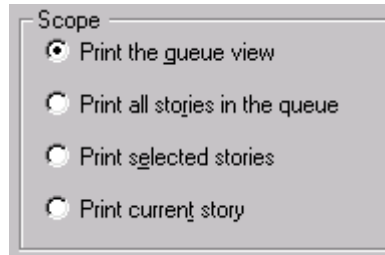
## Local Printing Dialog Box



The dialog box is divided into five sections: Scope, Options, Story Preview, Copies, and Grid. There are also several buttons, which include: Print Preview, Network, and Default. Each of these sections and buttons are described in more detail in this part of the chapter.

## Scope

The Scope section allows users to select one of four radio buttons to indicate what they want to print.



The options these four radio buttons offer are explained as follows:

The “Print the queue view” option determines whether the Story Preview checkbox is available. See “Story Preview” on page 12-34 for more information.

- **Print the queue view**- This option is available in most cases. In previous releases of Avstar NRCS, a user had to click in the Queue panel before using the Local Print dialog box to print the queue view. This is no longer necessary. For instance, if a user’s cursor is located in the Story panel, the user can still print the queue view by selecting this option.
- **Print all stories in the queue** - This option is available from the Directory panel or Queue panel if the queue has the printable (+P) database trait. This option will send the text of all stories in the queue to the printer.
- **Print selected stories** - This option is only available if a row or rows of stories are selected in the Queue panel. The option will send text of stories highlighted in the queue to the printer.
- **Print current story** - This option is available in all three panels of the Avstar Workspace. It will send the story that currently appears in the Story panel to the printer.

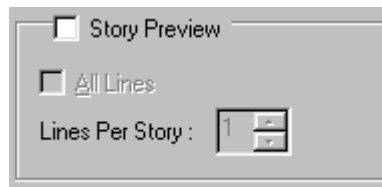
Default Option Selection:

- From the Directory panel, if the queue has the printable database trait, it defaults to “Print all stories.” If the queue does not have the printable database trait, it will default to “Print current story.”

- From the Queue panel with one or more stories selected by the row selector button, it will default to “Print selected stories.”
- From the Queue panel with no stories selected, it will default to “Print current story.”
- From the Story Panel, it will default to “Print current story.”

### Story Preview

The Story Preview section allows a user to print a preview or sample of each script as part of the queue image. Consequently, this section is only enabled if the “Print the queue view” option is chosen in the Scope section of the dialog box. Otherwise, it will appear gray and be unavailable.



The Story Preview section, once enabled, has three options a user can define, depending on how the queue view should be printed.

- **Story Preview** - When this check box is selected, Avstar NRCS will print lines of each story, along with the queue view, as defined by the user in the Story Preview section.
- **All lines** - When this check box is selected, Avstar NRCS will print all of lines of each story, along with the queue image. When this option is selected, the remaining option called Lines Per Story appears gray and is unavailable.
- **Lines Per Story** - When enabled, Avstar NRCS will print the specified number of lines of each story, along with the queue view. Lines of the story will be printed below the row that corresponds to it in the queue view. This option is disabled (appears gray) when the All lines option is checked.

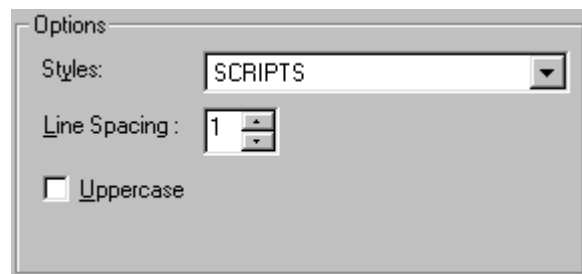
For instance, if a user wants to see the first few lines of each story in addition to a show's lineup, the user selects the "Print the queue view" radio button in the Scope section of the dialog box, selects the Story Preview check box, sets the Lines Per Story setting to 3, and clicks OK to print it. Avstar NRCS will print the first row of the lineup (or fields of the Story Form), followed by three lines of that story, then the second row and three lines of the second story, and so forth.



*Rows as shown in the Queue panel may not match exactly rows of the queue image printed with Story Preview selected. That is because Avstar NRCS' Story Preview printing feature is configurable. System administrators can define what information appears in the columns of a Queue panel and in fields of the Story Form panel. These two displays may or may not match exactly. This is also the case for Story Preview printing. For instance, a system administrator can designate which fields from the Story Form are not printed. The default is to print all fields in the Story Form, as specified within each story.*

## Options

The Options section of the dialog box contains three choices for users: Styles, Line Spacing, and Uppercase.



- **Styles** - A user can choose from a drop-down list of predefined styles. This feature is available when printing to both local and system printers.



*Various print style options can be defined by the system administrator and are stored in the first story in the queues within `SYSTEM.STYLES`. See "Local Print Style Options" on page 12-38 for more information.*

- **Line Spacing** - A user can configure how much space appears between lines of text. A setting of one would result in single-spaced text, two would be double-spaced, and so forth. This option will override any line spacing predefined in the chosen print style.



*The maximum number for line spacing allowed by default in the dialog box is 10. A print style can exceed this maximum and that setting will appear in the Line Spacing box when that style is chosen. However, if a user chooses to manually override the style setting by changing it in the Local Printing dialog box, then the default maximum of 10 will again take effect.*

- **Uppercase** - When this check box is selected, Avstar NRCS will print all text in uppercase (all capitalized letters). Whether this check box appears selected by default when the dialog box first opens can be set by the system administrator using a print style option. See “Local Print Style Options” on page 12-38 for more information.

### Copies



The Copies section allows the user to determine the number of copies printed and whether multiple pages are collated. The Collate check box is not available for selection unless multiple copies are printed—that is, the number of copies is set to a number more than 1.



## Grid

The Grid section of the dialog box contains two choices: Horizontal and Vertical.



- **Horizontal** - When this check box is selected, Avstar NRCS will include horizontal grid lines when printing the queue view image.
- **Vertical** - When this check box is selected, Avstar NRCS will include vertical grid lines when printing the queue view image.

The default behavior of these check boxes is based on a user's preferences.

## Print Preview and Network buttons

- **Print Preview** - This button allows users to preview queues or stories on their screens prior to printing.



*The Print Preview button is disabled when Story Preview is selected.*

- **Network** - This button offers the user more flexibility in selecting which printer is used. The dialog box has two locations from which a user can select a printer. The first is a drop-down list at the top of the dialog box that displays a list of printers loaded on, or locally connected to, the computer the user is on. The second is the Network button, which allows the user to select any printer available through the network.
- **Default** - This button returns settings of the Print dialog box back to default preferences of the user.

## Local Print Style Options

Avstar’s local print style options are added to the styles already used by Avstar NRCS for system printing. They are stored in the first story in queues within `SYSTEM.STYLES` with options for system printing. See “Creating a Style Story” on page 12-24 for more information.



*Do not make separate stories in the queues for system and local printing. Because all options for local printing begin with the letters WIN, Avstar NRCS will ignore these options when conducting a system print.*

In the story, options are specified with an option name—also known as a token—followed by an equal sign (=) and then the parameter value. Each option must be on its own line. A semicolon (;) will cause the remainder of that line to be ignored, and can be used to make notes in reference to the print option. The format looks like this:

```
<option name>=<parameter value>;<notes>
```

A sample story containing local print styles is provided later in this document. See “Example Style Story” on page 12-46 for more information.

The local print options are not case-sensitive, but do have default settings provided in the following table:

**Table 12-7 Local Print Style Options and Parameters**

Options	Parameter Values	Defined
WinPageFooter	<queue name>.<footer name>	Specifies the use of a footer. The <footer name> must match the name of a story in a queue, as defined by the <queue name>, located in <code>SYSTEM.STYLES</code> . Default is NONE.

Options	Parameter Values	Defined
WinPageHeader	<queue name>.<header name>	Specifies use of a header. The <header name> must match the name of a story in a queue, as defined by the <queue name>, located in SYSTEM.STYLES. Default is NONE.
WinHeaderFontFace	<font>	Specifies font used in the header and/or footer on the printout. Underscore <i>cannot</i> be used to represent a space. Does not recognize bold, italic, or underline. Default is MS Sans Serif.
WinHeaderFontSize	<font size>	Specifies a numeric value for the font size, in points, used in the header and/or footer on the printout. Default is 10.
WinBanner	YES   NO	Specifies whether to include the banner in printout. Default is yes.
WinLeftBanner	<format>	Defines banner format options (such as user name, page number, and queue name for printout) on left side of banner. See “Banner Format Options” on page 12-44 for more information.

Options	Parameter Values	Defined
WinRightBanner	<format>	Defines banner format options (such as user name, page number, and queue name for printout) on right side of banner. See “Banner Format Options” on page 12-44 for more information.
WinBannerFontFace	<font>	Specifies font used in the banner on the printout. Underscore <i>cannot</i> be used to represent a space. Does not recognize bold, italic, or underline. Default is MS Sans Serif.
WinBannerFontSize	<font size>	Specifies a numeric value for the font size, in points, used in the banner on the printout. Default is 10.
WinForm	YES   NO   ALL	Specifies whether a form is included in the printout of each story. The YES option specifies printing the form on first page of each story; NO specifies none; ALL specifies printing the form on every page of each story.
WinPrintForm	<form>	Defines which form is used for printing. Default is form specified within each story.

---

Options	Parameter Values	Defined
WinFormFontFace	<font>	Specifies font used in the form on the printout. Underscore <i>cannot</i> be used to represent a space. Does not recognize bold, italic, or underlined text. Default is MS Sans Serif.
WinFormFontSize	<font size>	Specifies a numeric value for the font size, in points, used in the form on the printout. Default is 10.
WinSeparator	YES   NO	Specifies whether a horizontal rule (line) is printed after the Story Form row and before the line of text from the story. Default is NO.
WinBlanks	<number of blank lines>	Specifies a numeric value for blank lines printed following each story when printing the Story Preview. Default is 2.
WinSpacing	<number of line spacing>	Specifies a numeric value for the line spacing of printed stories and queues. Default is 1, which results in a single-spaced printout.

---

---

Options	Parameter Values	Defined
WinScriptShift	YES   NO	Specifies whether Avstar NRCS selects the Uppercase checkbox in Local Printing dialog box by default, and prints all text in uppercase. Default is YES.
WinBodyFontFace	<font>	Specifies font used in body of the printout. Underscore <i>cannot</i> be used to represent a space. System will print bold, italic, or underlined text, as indicated in the original text of story. This option also applies to printed production cues. Default is MS Sans Serif.
WinBodyFontSize	<font size>	Specifies a numeric value for the font size, in points, used in the body of the printout. Also applies to printed production cues. Default is 10.
WinOrientation	<orientation>	Specifies an override of page orientation in the Printer Properties dialog. Allowable options are <code>portrait</code> or <code>landscape</code> . Default is as specified by the Printer Properties dialog.

---

<b>Options</b>	<b>Parameter Values</b>	<b>Defined</b>
WinScriptLeftWidth	<left width>	Specifies a numeric value for the width—measured in inches—of the left side of the story that contains production cues. Default is 3.
WinScriptRightWidth	<right width>	Specifies a numeric value for the width—measured in inches—of the right side of the story that contains story text. Default is 4.
WinScriptLeftMargin	<left margin>	Specifies a numeric value for the left print margin—measured in inches. Default is 1.25.
WinScriptRightMargin	<right margin>	Specifies a numeric value for the right print margin—measured in inches. Default is 1.25.
WinScriptTopMargin	<top margin>	Specifies a numeric value for the top print margin—measured in inches. Default is 1.0.
WinScriptBottomMargin	<bottom margin>	Specifies a numeric value for the bottom print margin—measured in inches. Default is 1.0.

## Banner Format Options

In Local printing, Avstar NRCS will include a banner at the top of printed text if the style selected by the user includes the `WinBanner=yes` option. The banner comes in two parts—a left side and a right side—each with its own user-configurable option. The system administrator can set what information appears in either side of the banner by using the local printing style options `WinLeftBanner` and `WinRightBanner`. These options recognize parameters only in certain formats, as shown in the following table:

Format Parameter	Defined
<code>&amp;u-name</code>	user name
<code>&amp;q-name</code>	queue name
<code>&amp;date</code>	date (according to Windows regional setting)
<code>&amp;time</code>	time (according to Windows regional setting)
<code>&amp;f-title</code>	slug
<code>&amp;f-page-number</code>	page number



*Literal text can be used with `WinLeftBanner` and `WinRightBanner` parameters. For instance, if you want the banner to print the word, “User” followed by a user’s name, define the parameter this way:*

`WinLeftBanner=USER-&u-name`

*See “Example Style Story” on page 12-46 for other literal text examples.*

To define a banner in a local print style, do the following:

1. Navigate to `SYSTEM.STYLES` in the Directory panel.
2. Double-click on the queue you want to modify. This will open or load that queue into the Queue panel.



3. Double-click on the first story in the queue. This will open or load that story into the Story panel.
4. Position your cursor in the Story panel.
5. Press Enter to make certain your cursor is at the start of a new line.
6. Type **WinBanner=yes** on the new line. If this line already appears somewhere in the style, you can skip step 7.
7. Type **WinLeftBanner=<format parameter>**. Separate multiple format parameters with single spaces. Format parameters are shown in the preceding table. For instance, if the left side of the banner should include the user's name and date, the local print style option would look something like this:

```
WinLeftBanner=&u-name &date
```



*The left banner print style option should be defined even if it will be blank; otherwise, the system uses the default setting. To define the left banner as blank, type: **WinLeftBanner=** leaving the remainder of the line after the equal sign blank. The left banner default setting is:*

```
WinLeftBanner=&u-name &date &time
```

8. Press Enter to make certain your cursor is at the start of a new line.
9. Type **WinRightBanner=<format parameter>**. Separate multiple format parameters with single spaces. Format parameters are shown in the preceding table. For instance, if the right side of the banner should include the story slug and queue name, the local print style option would look something like this:

```
WinRightBanner=&f-title &q-name
```



*The right banner print style option should be defined even if it will be blank; otherwise, the system uses the default setting. To define the right banner as blank, type: **WinRightBanner=** leaving the remainder of the line after the equal sign blank. The right banner default setting is:*

```
WinRightBanner=&q-name &f-title &f-page-number
```

10. Save the story.

## Example Style Story

Here is an example of a local print style story stored in a queue located in SYSTEM.STYLES:

```
WinBanner=yes
WinLeftBanner=&q-name PAGE-&f-page-number SLUG-&f-title
WinRightBanner=USER-&u-name &date &time
WinBlanks=2
WinScriptShift=No
WinScriptLeftWidth=3.5
WinScriptRightWidth=3.5
WinLeftMargin=0.75
WinRightMargin=0.75
WinTopMargin=1.0
WinBottomMargin=1.0
WinSeparator=Yes
WinSpacing=1
WinForm=All ;NOTE-Other options=Yes, No
WinPrintForm=Rundown
WinHeaderFontFace=Times New Roman
WinHeaderFontSize=10
WinBannerFontFace=Arial
WinBannerFontSize=12
WinFormFontFace=Arial
WinFormFontSize=10
WinBodyFontFace=Times New Roman
WinBodyFontSize=20
WinOrientation=portrait;Note-Only other options=landscape
```

The example style story shows only those style options pertaining to local printing. An actual style story may also contain style options pertaining to system printing. System style options are ignored when Avstar NRCS handles a local print job.

---

## Managing Printers

When someone prints a single story or an entire queue from the workstation, it is called a *print request*. Your system allows any number of users to send print requests to the same printer at the same time. Your system schedules print requests in the order it receives them.

When you add a printer to your system, the system administrator creates a print queue for that printer in `SYSTEM.PRINTERS`. The queue must not be inverted to hold requests in the proper order. For instance, the queue for printer 3 is called `SYSTEM.PRINTERS.003`.

Each print request is identified by its title (or queue), the time the user printed it, user name, style used, number of copies to be printed, and `print` command used.

Your system places the first print request it receives for a printer in that printer's print queue and then sends it to the printer. This is called the *current print request*.

As more requests come in, they are placed below the current request in the order received. These requests are called *pending print requests*. When the printer finishes printing the current print request, the system removes it from the queue and moves the pending print request below it up, making it the new current print request.

### Removing a Pending Print Request

Pending print requests are held in print queues like stories. You can remove any pending print request that you do not want printed.

To remove a pending print request:

1. Open the appropriate print queue under `SYSTEM.PRINTERS`.
2. Select any print request.
3. Click the Edit drop-down menu.
4. Select Kill.

## Restarting the Current Print Request

To rebuild the print job:

1. Fix whatever caused your print job to be canceled.
2. Restart the printer using the `restart` console command.

For instance, if printer 35 jams, fix the jam and then rebuild the print job by typing:

```
restart 35
```

## Reordering a Pending Print Request

You can rearrange pending print requests by reordering the stories in `SYSTEM.PRINTERS`. The printer does not stop while you reorder the queue. If the current print request finishes while you are reordering the queue, the next story at the top of the queue is processed.

## Cancelling a Runaway Print Job

You can cancel an unwanted print job, such as printing the entire contents of a queue instead of a single story. To cancel it, do the following:

1. Turn off the printer or put it offline.
2. Navigate to the `SYSTEM.PRINTERS` queue for the printer at an Avstar Workstation.
3. Kill the job request.
4. Restart the printer port number at the Avstar console.
5. Turn the printer back on, or put it online.

## Responding to a “Printer Offline” Problem

Occasionally, a system printer to which a print request has been submitted is in an offline condition. The user who submitted the print request receives the following message:

```
[System] P199: Unable to continue printing
```

The number following P is the device number of the printer to which the request was submitted. A similar message appears on the Avstar console when the offline condition is first detected.

A printer is considered offline by Avstar NRCS if it has not responded to normal flow control for three minutes. Consult your printer hardware manual for information on recovering from an offline condition.



# CHAPTER 13

## *Wires*

As explained in Chapter 11, “Configuration Files,” information about all hardware and software components on your Avstar NRCS system is contained in profile files. Therefore, each wire service available to your system has a wire profile file. This file contains commands your system needs to control wires, and also contains settings for options you can use with that wire service. When you restart a wire service, your system checks its profile for necessary information.

Major sections of this chapter are:

- Adding a Wire to Your Avstar System
- Wires Profile Files
- Wire Profile Options
- Using Special Characters in a Profile
- Distributing Stories from the Wire
- Operating Wire Keyword Searches
- Keyword Checker Messages

## Adding a Wire to Your Avstar System

To add a wire service to your system, you must complete four phases in the setup.

- Phase 1 - Connect the wire service to a PCU port.
- Phase 2 - Create an entry for the wire service in the configuration file and a wire profile for the wire service.
- Phase 3 - Add the wire distribution information.
- Phase 4 - Restart the wire service's program to incorporate changes you made, and so it can begin receiving and distributing wire stories.

### Phase 1

To connect a wire service to Avstar NRCS, do the following:

1. Choose a device number for the wire service.

Find an unoccupied PCU port. For instance, an AP wire is added to port 6 on PCU 10, which means the wire has device number 16. See "PCU Device Numbering" on page 11-34 for more information.

2. Connect wire to selected PCU port.

### Phase 2

Wire service information must be added to the configuration file on *each* server of your system. That includes adding the wire service's device number to the PCU's configuration line and adding a configuration line for the wire program that processes stories sent by the wire service. See "Changing the Configuration File" on page 11-15 for more information.





*If you move a wire service to another PCU, you must modify the entry for the wire service's program. Move the wire service's device number from the old PCU's configuration line and add it to the new PCU's configuration line. Give the wire service a new device number that reflects the new PCU port to which you attached it.*

To modify the configuration file for a wire service, do the following:

1. Select *all* servers. See “Selecting Servers” on page 2-5 for more information.

2. Type:

```
AVSTAR-A: ed /site/config
editing /site/config
1259
```

3. Modify the configuration file (`/site/config`), by doing the following:

- a. Add the wire's device number in the position that represents the port to which you have connected the wire service. For instance, for the wire service connected to port 6 of PCU 10, the wire program's device number (16) is added to PCU 10's configuration line. At first, the line may appear similar to this:

```
ccu 10 pcu10 at 11 12 13 14 15 - - -
```

Ensure that you place the new device number in the position representing the port to which you connected the wire. For instance—for the wire with device number 16—the wire's device number is placed in position 6 in PCU 10's configuration line. The new line should look like this:

```
ccu 10 pcu10 at 11 12 13 14 15 16 - -
```

- b. Add a configuration line for the new wire service. By convention, the configuration lines are arranged according to device number. For instance, the configuration line for the wire service with device number 16 would be placed after the configuration line for device 15.

This procedure, which modifies the `/site/config` file uses `ed`, the UNIX line editor. If you do not know how to use `ed` to modify lines in the file as required in step 3, please see “Using the UNIX Line Editor” on page 10-4.

A wire program's configuration line uses this format:

```
wire <dev #> <speed> <program> <source> <join option>
```

Parameter	Description
device #	The wire's device number. Include this device number in device numbers list in the configuration line for the PCU.
speed	Sets baud rate (bits per second) at which the wire communicates with the PCU.
program	Used to process the wire. Ask the wire provider which communications standard the wire conforms to.
source	The wire's source name (capitalized portion of the distribution name—AP, UP, and so forth). It is the same as the source name used in the wire distribution story. The device name for a wire program is the source. Use the source name to create an entry in the search job list for the wire program.
join option	Merges consecutive stories that have the same title. It is used with wire services that routinely break stories into multiple takes. If you place <code>join</code> in this position, users can view all takes of a story with the <code>get all</code> or <code>read all</code> workstation commands. To leave this option out of the configuration line, enter a hyphen.

For instance, the wire program device number is 16, and it needs to communicate with the wire service at a speed of 1200 baud, 7 bit, even parity. Also, the wire program uses the `anpa7` program, and will be referred to as AP. So, AP is the wire's source. The new device configuration line looks like this:

```
wire 16 1200-7e anpa7 AP - ; ap wire
```

Do *not* use an uppercase (W) in step 4. See “ed, the Line Editor” on page 10-1 for more information.

4. When you finish making changes to the configuration file, you must save your changes. To do this, type:

```
w
```

After you press enter, a number will appear representing the file size, such as:

```
1306
```

5. Exit the UNIX line editor by typing:

```
q
```

6. Create a profile for the wire.

The easiest way to do this is to copy the standard profile for the type of wire service the wire program works with to the file holding that wire program's profile, then modify the copy.

A wire program's profile must be in a file with the same name as the wire service's device number. That file must be located in the `/site/wires` directory.

For instance, an AP wire service is installed at port 6 on PCU 10, and given device number 16. Accordingly, the profile for the AP wire service is copied to a new file called `/site/wires/16`. After the profile is copied, modifications can be made.



*If you move a wire service to a new PCU port and give it a new device number, you need to give the corresponding wire program's profile a new name that matches its new device number. The best way to do this is to use the `mv` (move) command. Type `mv` followed by the pathname of the file you want to change to the new pathname you want the file to have.*

To copy and modify a wire profile, do the following at the console:

1. Select *all* servers.
2. Use the `cp` command to copy the profile for the wire service. For instance, to copy `/site/wires/anpa7` (the profile for an AP wire service) to a file on each server called `/site/wires/16`, type:

```
cp /site/wires/anpa7 /site/wires/16
```

3. To determine whether you need to make changes or additions, use the `cat` command to view the file. The format is:

```
cat <file pathname>
```

For instance, to examine the file for the wire service connected to port 6 on PCU 10, type:

```
cat /site/wires/16
```

A message similar to the following will appear:

```
; @(#) release/wires/anpa7      11.1      Date 00/05/22
; anpa7 - suitable for upi7, ap7, and cupi7
bits          7
start         <soh>
eos          <EOT>
category     -----
flags        peosline
title       30
;
map          <16>    <SP>                ; TTS SPACE BAND
...
```



*Before you modify a profile, use the `cp` command to make a backup copy of the wire profile. Use the backup copy to restore the file.*

*To back up the profile, type `cp` followed by the full pathname of the file you want to copy and the full pathname of the file to which you want to copy it. (The destination filename can be a new file.)*

4. If changes are needed, then edit the profile by doing the following:
  - a. Select *all* servers. See “Selecting Servers” on page 2-5 for more information.

b. Type:

```
AVSTAR-A: ed /site/wires/16
editing /site/wires/16
245
```

- c. Navigate to and modify lines in the file, as necessary. For instance, to have the wire program convert all text in each story it processes to uppercase, the word `uppercase` should be added after the `flags` option.

At first, the line may look like this:

```
flags    peosline
```

This procedure, which modifies the `/site/wires/16` file uses Ed, the UNIX line editor. If you do not know how to use Ed to modify lines in the file as required in step 7, please see “Using the UNIX Line Editor” on page 10-4.

The line should now look like this:

```
flags  peosline  uppercase
```

Do *not* use an uppercase (w) in step 4. See “ed, the Line Editor” on page 10-1 for more information.

- d. When you finish making changes to the file, you must save your changes. To do this, type:

```
w
```

After you press enter, a number will appear representing the file size, such as:

```
306
```

- e. Exit the UNIX line editor by typing:

```
q
```

### Phase 3

Each wire program must have its own set of instructions in a story, called the *wire distribution story*. When you add a new wire to your system, you need to add new instructions to your system’s wire distribution story so it knows how to distribute stories it receives from that wire service.

The wire distribution story is located in the Distribution queue in the `SYSTEM.WIRES` directory.

To view the story, log in as a system administrator at an Avstar Workstation, and navigate to `SYSTEM.WIRES.DISTRIBUTION`.

Here is a partial sample of a wire distribution story:

AP#####	wires.all	always	URGENT
AP#####f#	wires.biz		URGENT
AP#####a#	wires.national		URGENT
AP#####n#	wires.national		URGENT
AP#####w#	wires.washington		URGENT
AP#####i#	wires.world		URGENT
AP#####s#	wires.sports		
AP#####o#	wires.weather		
APb###aa##	wires.features		
AP#kip01##	wires.kipling		
AP#bloom##	wires.bloomberg		



*If you move a wire to a different PCU, the system continues to use whatever instructions are in the wire distribution story for that wire.*

Before making any modifications to the wire distribution story, make a backup copy of the story.

To modify the story, do the following at an Avstar Workstation:

1. Select the story in the Queue panel. The story's text will appear in the Story panel.
2. Click in the Story panel to place your cursor there.
3. Edit the story by changing or adding lines, as necessary.
4. Save the story, by doing one of the following:
  - a. Click the save button.
  - OR-
  - b. Click the File drop-down menu and select Save.



*The wire distribution story is typically set up by iNews Customer Support technicians when your system is installed. If you need assistance in modifying an existing wire distribution story, contact iNews Customer Support. Also, a standard copy of a wire distribution story is available in Appendix B, “System Files.”*

## Phase 4

The final phase in adding a wire to your system is to check your setup for errors and restart the PCU and wire service. To do this, do the following:

5. (Optional) Test your configuration changes for errors by using:

```
configure /site/config <system> <computer>
```

For instance, to test configuration changes for a wire added to a PCU running on server A in an AB system, you would type at the console:

```
configure /site/config ab a
```

When the console prompt returns, the configuration file has been checked. If the system detects any errors, it displays appropriate bad configuration messages.

6. Do one of the following:
  - a. If you modified *only* an existing wire program’s profile and did not make any changes to the configuration file, you do not need to reconfigure the system. Restart the wire and it begins to use changes you made to its profile. To do this:
    1. Select the server to which the PCU is connected.
    2. Use the `restart` command in this format:

```
restart <device number>
```

For instance, to restart the device at port 6 of PCU 10, type:

```
AVSTAR-A: restart 16
```

```
AVSTAR-A: P16: 12:43:20 Hot-to-go
```

When you see a `Hot-to-go` message for the wire service, as shown in the previous example, you can begin to use it. If there are any errors in the profile, the system reports them on the console when you restart the wire service.

-OR-

- b. To have the system adopt changes you have made to the configuration file, reconfigure the system by doing the following:
  1. Log out the PCU. For instance, to log out PCU 10, type:  
`logout 10`
  2. Stop the PCU. For instance, to stop PCU 10, type:  
`stop 10`
  3. Select the master computer (typically server A). See “Selecting Servers” on page 2-5 for more information.
  4. Become a console superuser. See “Becoming a Console Superuser” on page 3-2 for more information.
  5. Take the system offline. Type:  
`AVSTAR-A# offline`
  6. Reconfigure the system. Type:  
`AVSTAR-A# configure`
  7. When the prompt returns, use `online` to bring the system back online again. Type:  
`AVSTAR-A# online`
  8. Exit from superuser mode. (CTRL-D)

A message similar to the following will appear:

```
A Fri Oct 6 00:18:58 2000 msg System being configured
```

9. When you see the `System being configured` message, select the server to which the PCU—and wire ser-



vice—is connected and use the restart command. For instance, to restart PCU 10, type:

```
AVSTAR-A: restart 10
```

You will see `Hot-to-go` messages indicating that the devices on the PCU have been restarted and are ready to use.

10. (Optional) Back up site files using the `sitedump` command.

## Wires Profile Files

Each wire program's profile contains instructions that tell the program how to interpret and process stories it receives. Your system must have a separate wire program for each wire service connected to it. Each wire program, in turn, must have its own wire profile to process stories sent by the wire service.

Wire profiles are text files located in the `/site/wires` directory. Your system is installed with standard profiles for most wire services. To see which wire profiles are already installed on your system, use the `ls` command to produce a list of filenames. Type:

```
AVSTAR-A: ls /site/wires
```

Information similar to the following appears:

77	apa6	bbcgn5	europa5	notimex5	sports7	ukpa5
81	apr5	bbcmon5	globo5	prime7	sticker7	ukpa6
aap7	aptv7	busines7	gns5	prwire7	tass7	ukreut5
abc7	asca5	cbs7	gns7	radcor5	tndm7	ukreut7
anpa7	baycity7	city7	iptc7	reut5	ukafp5	uktelex5

This list contains numbered filenames, such as 77 and 81, and alphanumeric filenames, such as `anpa7` and `aptv`. The alphanumeric files are the standard profiles for several different types of wires. For instance, the `anpa7` file is the standard profile used by wires such as AP and UPI.

The numbered files are profiles your system actually uses. These files correspond to device numbers used by your system's wire programs. Device numbers are specified in the configuration file. Each wire program looks for its profile in a file in the `/site/wires` directory. The filename must match the wire program's device number. For instance, a wire program with device number 17 looks for its profile in a file named `/site/wires/17`.

To see what a wire profile looks like, use the `cat` command to list the standard wire profile:

```
AVSTAR-A: cat /site/wires/anpa7
```

A display similar to the following appears:

```
; @(#) release/wires/anpa7      11.1      Date 89/05/22
; anpa7 - suitable for upi7, ap7, and cupi7
;
bits          7
start         <soh>
eos          <EOT>
category     -----
flags        peosline
title        30
;
map <16> <SP>          ; TTS SPACE BAND
map <25> <SP>          ; EM SPACE
map <29> <SP>          ; THIN SPACE
map <30> <SP>          ; EN SPACE
...
map <124> 5/8
map <125> 3/4
map <126> 7/8
```

## Wire Profile Options

In general, each wire option available may appear only once in a wire profile and may be assigned only one value. Exceptions are noted in the descriptions.

`accent <character>`

To convert wire codes into accented characters, a code is used to identify the accent sequences. To enable accent sequence conversions, specify the code that the wire service sends with this command. No accent conversions are done if this command is omitted. Typically a backspace (<bs>) is used to identify accent sequences.

`answer <code string> <name string>`

For wires that require an answerback string sent before a story is received, include a line with `answ` followed by the code string sent by the wire service and the name string the Avstar Newsroom Computer System answers back.

`bell <character>`

Some wire services send bell codes, which cause Teletype machines to ring a bell, to indicate the relative priority of a story. The more bell codes sent with a story, the more urgent the story. Usually, only the older, slower wire services do this.

Although your wire program does not assign a relative priority to a story based on the number of bell codes in the story, it can upgrade a story from routine (r) to urgent (u) if the story contains a bell character. Type `bell` followed by the character the wire service uses as a bell code.

This upgrades a routine story to urgent; it does not do anything if the story's priority is not set to routine. The bell character can be any character.

`bits [5 | 6 | 7 | 8 | ibm]`

Each character sent by a wire service is represented by a number of data bits (ones and zeros). Different wire services use different

numbers of data bits to represent the characters they send. Older, slower wire services use as few as five bits; newer, faster wire services use seven or eight bits.

See “IBM Character Set” on page E-8 for more information.

You must use the `bits` option in each wire program’s profile to specify the number of bits the wire service uses to represent characters it sends. Type `bits` followed by the number of bits the wire service uses. Take special care not to specify an incorrect number, because that may make it impossible for the wire program to receive stories transmitted by the wire service. Type `ibm` as the bits value for a wire that sends IBM PC format 8-bit characters.

The number of bits you specify also determines the character translation table the wire program uses. This table tells the wire program how to convert decimal values it receives to characters. The number of bits you specify also sets the stop bits, parity, and handshake options to values normally used by wire services that use that number of bits.

Table 13-1 shows the character table, stop bits, parity, and handshaking selected when you specify a certain number of data bits.

See “ASCII (7-bit) Character Set” on page E-2 for more information

**Table 13-1 Bits in Wire Services**

Bits	Character Table	Stop Bits	Parity	Handshake
5	Baudot	1.5	None	None
6	TTS	1	None	None
7	ASCII	1	Both	None
8	ASCII+DEC MCS	1	None	None
ibm	ASCII+PC	1	None	None

You can change stop bits, parity, and handshaking values individually using the `stopbits`, `parity`, and `handshake` keywords within the wire profile.

```
category <string>
```

Each wire story has a distribution code that consists of the wire program's source name, the story's category codes, and the story's priority code. Since different wire services attach different numbers of category codes to their stories, use the `category` keyword to specify how many category codes the wire program can expect with each story it receives from the wire service.

Follow the keyword with a number of printable characters equal to the number of codes the wire sends. For instance, if the wire service sends seven category codes, follow `category` with seven hyphens as shown here. Hyphens are used in standard profiles by convention, but you can use any printable character.

```
category -----
```

Type the characters instead of the actual number to allow the wire program to pad the category codes with those characters if the number of codes sent is smaller than what you specify. For instance, if you use the `category` keyword as shown and the wire program receives a story with only four category codes, it pads the unused positions with hyphen (-) characters, such as `an--j-x`.

The total distribution name cannot be longer than 10 characters. Ensure the category length you specify combined with the length of the wire's source name and length of the story's priority code is no more than 10 characters. A wire program's source name is defined in its configuration line. Priority codes are always one-character long.

```
distribution <category> <word>
```

If the story contains a certain word, two wire programs—`abc7` and `ukpa6`—can assign a category code to a story. Follow the `distribution` keyword with the category you want to assign, and then the word the story must contain to be assigned that category. For instance, to assign the category `xlf` to stories that contain the word `moonshot`, type:

```
distribution xlf moonshot
```

The wire program searches only for a whole word. It does not recognize the word if it is part of a larger word. You can specify multiple distributions. Order of `distribution` lines in the profile is significant, because the system searches for words in the order they are given and stops searching a story after the first match, ignoring any subsequent distribution lines.

Values you supply for the category and word must consist of printable characters. The distribution keyword is used by two wire programs—`abc7` and `ukpa6`—all other programs ignore this keyword if you use it in their profiles. If the string you specify contains a nonprintable character, use angle brackets (`<` `>`) around it.

```
end <string>
```

If the wire program receives stories from a wire service that marks the end of each story with a special string, use the `end` keyword to define that string in the program's profile. Some wire services mark the end of each story with a special character instead of a string. In that case, you can use the end-of-story (`eos`) keyword to define the character.

To define the end-of-story string, type `end` followed by the string the wire service uses. This string must consist of printable characters. For instance, if the wire service uses `nnnn` as the end-of-story string, in the wire profile type:

```
end      nnnn
```

A wire service may alternate between using an end-of-story character and an end-of-story string. In that case, you may include both `end` and `eos` in your profile. The wire program considers either to indicate the end of the story.

```
eol <character>
```

If a wire service uses a special character (other than the standard line feed) to indicate the end of a line, use the `eol` option to define that character so it is converted to a line feed in all stories received from that wire service.

```
eop <character>
```

Some wire services use a special character to indicate the end of a paragraph in a story. To display these stories properly, you must convert that end-of-paragraph character to a standard HEOL (Hard-End-Of-Line) character. To do this, use the `eop` option to define the character that wire service uses as the end of paragraph marker. The wire program only converts the `eop` character when it appears as the last printable character on the line.

For instance, if the wire service uses the greater than sign (>) to indicate the end of a paragraph, you can use `eop` to convert any occurrence of a > at the end of a line to an HEOL. Type:

```
eop    >
eos    <character> <character> <character> . . .
<character>
```

If the wire program receives stories from a wire service that marks the end of each story with a special character, use the `eos` keyword to define that character in the program's profile. Some wire services mark the end of each story with a special string of characters. Use the `end` keyword to define the string.

While wire services that use an end-of-story character generally use the same character consistently, not all do. Some wire services change characters with each Teletype operator shift change. If this is the case, follow `eos` with all the different end-of-story characters the wire service uses. The wire program considers the first occurrence of any of these characters in a story to signal the end of the story.

To define the end-of-story character, type `eos` followed by the characters your wire service uses. These characters can be any character.

For instance, if the wire service uses `EOT` as the end of story character, enter the following in the wire profile:

```
eos    <EOT>
```

A wire service may alternate between using an end-of-story character and using an end-of-story string. You can include both `end`

and `eos` in your profile. The wire program considers either one to indicate the end of story.

```
flags [lock | char | pblines | peosline | uppercase  
| adtheol]
```

The `flags` keyword allows you to control certain aspects of how wire stories are displayed on your system. This keyword accepts up to five values, each value separated by a space.

If your system is receiving a wire that uses a case-shift character, you need to include a line in that wire's profile that translates this character to the value Avstar NRCS expects (decimal `<130>`). If the case-shift character from a particular wire service has a decimal value of `<0>`, the instruction looks like this:

```
map <0> <130>
```

For more information about the `map` instruction, see the `map` keyword.

- `lock`

Certain wires transmit only lowercase characters, but send a case-shift character to indicate when to start uppercase display. If you follow `flags` with `lock`, characters sent after the case-shift character in a wire story are all converted to uppercase until some other shift character is sent.

- `char`

Using `char` instead of `lock` in the `flags` line causes only the next character following the case-shift character to be converted to uppercase.

- `pblines`

A wire program usually removes blank lines, such as lines used to separate paragraphs, from the wire stories it processes. If you do not want these lines removed, follow `flags` with `pblines`, which prints blank lines.



- `peosline`

Normally, a wire program will not include the line containing the end-of-story string or end-of-story code in a wire story. If you follow `flags` with `peosline` (print end-of-story line), the wire program includes the line containing the end-of-story string or end-of-story code as part of the story text.

- `uppercase`

By default, a wire program leaves the text of each wire story as the wire service transmitted it. However, to force the wire program to shift text in each story to uppercase, follow `flags` with `uppercase`.

- `addheol`

Some wires do not provide enough formatting information for handling tabular material, which makes reading sports scores, weather tables, and stock listings difficult on an Avstar Workstation. The `addheol` option causes the server to treat each line break as an end of paragraph. The default wire profile condition is that there is no `addheol` handling done.

```
flash <priority character> <word>
```

A `flash` option line specifies a correspondence between a certain word that may appear in a received story and the priority code character to be assigned to any story that contains that word. For instance, to assign the priority `u` (urgent) to stories that contain the word `moonshot`, type:

```
flash    u    moonshot
```

The wire program searches for the full word; it does not recognize the word when it is part of a larger word. You can use up to 16 `flash` keywords in a profile. The order of `flash` lines in a profile is significant, because the wire program searches for words in the order they are given, and stops searching a story after finding the word currently searched.

You must supply printable values for both the priority character and the word. Also, while most wire programs search the entire

text of each story for the word, a few (ukpa5 and ukpa6) search only the title.

`form <name>`

The `form` option is the form name assigned to the wire story, generally used to view the story. This form is the story form assigned to queues that typically contain wire stories.

To ease the migration process from Avid NetStation to Avstar NRCS, the `template <number>` option is treated as a `form <name>` option, with the number converted to a three-digit number with leading zeroes. If a form name is not given in the profile, it will default to 100, and will expect to use a form in `system.forms.1.100`.

`handshake [auto | xon | both | none]`

The `handshake` keyword specifies the type of handshaking (flow control) used by the wire program. If you leave this option out of a wire profile, the wire program uses the default value, which is `none`. Since wire services do not usually use handshaking, most wire profiles do not include this option.

If the wire service uses handshaking, you must include `handshake` to specify the type it uses. If you follow `handshake` with `xon`, the wire program uses XON/XOFF software flow control. If you follow it with `auto`, the wire program uses RTS/CTS hardware flow control. If you follow it with `both`, the wire program uses both in parallel.

`idle <timeout> <interval> <notify group>`

The `idle` profile option controls how your system handles a prolonged period of inactivity, when no stories are received by Avstar NRCS, for a wire. The timeout value indicates how long the wire should be idle before users are notified. For instance, a time-out value of 2:00 triggers notification if no stories are received from this wire in two hours.

The `notify group` is optional; it contains the user group name on Avstar NRCS that is notified, through the top-of-screen message, if

the time-out value is reached. A message is always sent to the console when this happens. To disable sending a message to a group, put a hyphen (-) in this position.

The interval value is used to repeat this notification if the wire remains idle for an additional period of time beyond the time-out value. A time-out value of 2:00 combined with an interval of 15 notifies users after two hours of idle time and every 15 minutes thereafter, as long as the wire remains idle.

The time-out and interval can have any values between 2 minutes and 24 hours, expressed in `hh:mm` format. Providing a value of 0 for the time-out, or omitting the `idle` option from the profile altogether, disables the idle time alert for this wire. An interval of 0 limits notification to one initial message.

For more information, see Appendix E, “Character Mapping Code Tables for Wires.”

```
map <incoming character> <translated string>
```

A `map` keyword translates a particular received character code into one or more characters that can be displayed.

When you select the `bits` value, a standard translation table is selected. The `map` command allows you to override the standard translation table for individual characters.

Almost every wire service uses a few characters in a nonstandard way to represent such things as fractions and em/en spaces and hyphens, using variable character widths. Since each wire service may use different characters, they cannot be translated in the standard character-translation table. Instead, they must be translated individually using the `map` keyword.

To translate a character to a string that can be displayed, follow `map` with the wire service character and the character or string you want to translate it to. For instance, suppose the wire service uses a value of 91 to represent the fraction 1/2. To display this on a workstation, map 91 to the string 1/2.

```
map <91> 1/2
```

Enter the character you want to translate and the string to which you want to convert it as either decimal values in angle brackets,

such as <32>, or as actual characters, such as a. A space in the string ends the string. To translate to a string that contains a space, use <sp> to represent the space, such as a<sp>b.

You may use up to 64 `map` keywords in a profile. Decimal values used in `map` lines can be any value from 0 to 255.

`pagenumber <length>`

The `pagenumber` keyword specifies the number of characters allowed as PAGE-NUMBER field information. The default value is five, with the maximum number of 18 characters that can be collected. Most wire programs collect a `story number` from the wire input. The length of this information varies from wire to wire. The `pagenumber` keyword is similar to the `title <length>` and `presenter <length>` options.

`paragraph <number>`

Wire programs generally expect the first line in a new paragraph to begin with three to five leading spaces. When the wire program processes a story, it adjusts any line with at least that many spaces to the standard five-space indent.

However, some wire services use only one or two leading spaces to indicate when a line is the start of a new paragraph. If you have such a wire service, you must let its wire program know how many leading spaces indicate the beginning of a new paragraph. Use the `paragraph` keyword, followed by the number of leading spaces the wire service uses to indicate when a line is the start of a new paragraph.

You cannot set this option to a value greater than four.

`parity [odd | even | both | none]`

To set the `parity` option to something other than the default value assigned with the `bits` keyword, include a `parity` value of `odd`, `even`, `both`, or `none`, depending on whether the wire service includes a parity bit with each character it sends and, if so, the kind of parity it uses.

Set this option to `none` if the wire service does not use a parity bit. This is usually the default setting, so if you do not include the `parity` option in the profile, the wire program expects the wire service not to use a parity bit. Set it to `even` if the wire service uses even parity. If the wire service uses odd parity, set this option to `odd`. If the wire service includes a parity bit with each character it sends, but the bit's setting does not matter, set this option to `both`. This instructs the wire program to expect a parity bit but ignore its value.

```
presenter <length>
```

The `presenter` keyword specifies how many characters are allowed as PRESENTER field information. The default value is 0, which means that no `presenter` data will be stored in the wire story. Only wires using the `flash` option and the `bbcmon5` wire program collect data, which is stored in the PRESENTER field.

The `flash` word found in the story that causes the priority to change consists of the data stored in the PRESENTER field. If you specify any `flash` options, also specify a `presenter <length>` option. `Presenter` is similar to the `title <length>` keyword option.

```
priority <character>
```

If you have a wire service that frequently transmits wire stories without priority codes, use the `priority` keyword to assign a default priority code to each story you receive. This is usually used with the `flash` keyword, which assigns specific priorities to stories that meet certain criteria.

To set a default priority, type `priority` followed by a printable ASCII character indicating the priority you want attached to stories. Common values for this option include `r` for routine (the default), `u` for urgent, `f` for flash, and `b` for bulletin. If you do not specify a value, the default is `r`. If you do not want to attach a default priority to the stories, enter `priority <nul>`.

See "dbrestore Conversion Map" on page E-15 and "Sample Arabic Wire Profile" on page E-16 for more information.

```
reverse <characters> [<embedded characters>]
```

The wire program has the ability to reverse the order in which figures are displayed in the database, using the wire profile parameter called `reverse`. This parameter is typically used at sites where the written language is from right to left, such as Arabic sites. The syntax of this parameter includes:

- `<characters>` that are included in the sequence to be reversed. Any of these characters can signal the start or end of the sequence.
- `<embedded characters>` that are included in the sequence to be reversed; however, they cannot start or end the sequence. This string need not be present. An example would be:

```
reverse 0123456789 /,.
```

So any sequence of characters starting and ending with a digit that may include a period, comma, or slash will be reversed. Here are a few examples:

An input line the systems as 9991/70/60. would be converted to the systems as 06/07/1999.

The period was not included in the reversal since it is an embedded character and cannot be the last character in the sequence.

The line "we had \$00.000,01." is converted to "we had \$10,000.00."

The period between the zeros is reversed, along with the digits, since it is "embedded" in the sequence.

```
start <string>
```

The `start` option specifies the exact string of characters that indicates the start of a received story. The entire string must be found to start a story. The start string may include any character. You can

use the wildcard character <128> within the string to stand for any number of other characters.

```
stopbits [1 | 1.5 | 2]
```

To set the number of stop bits to a value other than the default, use the `stopbits` keyword followed by the number of stop bits the wire service transmits with each character. You generally do not need to use this option, because setting the `bits` option usually sets the appropriate `stopbits` value as well.

```
title <length>
```

The `title` keyword specifies how much of each story's title the wire program puts in the story's title field. The default value is 20 characters. If you want more or fewer characters, use `title` to specify how many. For instance, `title 30` causes the wire program to use the first 30 characters of each story's title.

While this option determines how much of a story's title is available for display, the size of the title field in the form where the story is sent controls how much of the title can be displayed. The length of the form fields is not specified in characters, and, because variable pitch fonts are used, the 30 characters may not fit in a field of 25 units.

Often, wire services transmit stories with titles longer than the title field. You can use the `title` keyword to have the wire program display the entire title at the top of each story, as well as in the title field. Set the title to a value equal to or less than the width of the title field. If a story has a title longer than this value, it displays as much as it can in the story's title field, and then displays the entire title at the top of the story as well.

If you want the wire program to display as much as it can in the story's title field and not display the title at the top of the story, specify a large value, such as 80, for this option. This will not affect the title field, because characters that do not fit into it are lost.

`tli <character>`

The `tli` option allows most wire programs (excluding `atelex5` and `baycity7`) to recognize lines of text, starting with a tab line indicator. If you do not specify a `tli` option, it defaults to a back-space character. Wire input conforming to `anpa` format has lines that contain tabular information and is introduced with a `tli`. The wire programs ensure that each of these lines is stored in the database as a separate paragraph. You can disable this feature by defining the `tli` to be zero (null).

## Using Special Characters in a Profile

Some characters cannot be produced from the keyboard or are interpreted differently by your system and, therefore, you must enter them in your profile in a special way. Characters such as semicolons (;), pound signs (#), and at (@) symbols have a special meaning for your system or your system's console. Additionally, you may need to set an option to a value that includes nonprinting characters, such as those produced by the Escape (ESC) and Enter keys.

## Entering Nonprinting Characters

To add a nonprinting character to a profile, you must use either the alias or the decimal value.

### Entering Characters by Alias

Your system recognizes a number of aliases for many of the nonprinting characters you may use in a wire profile. These aliases are two or three-letter abbreviations that represent different characters. For instance, `esc` represents the Escape key and `cr` represents the Enter key.



When you put an alias in a wire profile, you must enclose it in angle brackets (< >) so your system knows that it is an alias. For instance, here is an `eos` line that uses the Escape character:

```
eos      <esc>
```

## Entering Characters by Decimal Value

You can also represent any character, nonprinting or otherwise, by its decimal value. Enter the character's decimal value in angle brackets where you want it to appear in the profile.

For instance, Escape has a decimal value of 27, and Enter has a decimal value of 13. Here is an example of an `eos` line using the `esc` character:

```
eos      <27>
```

## Avoiding Characters Used by the System

Do not include any of the following characters as control codes. The characters, and how the system interprets them, are listed in Table 13-2.

**Table 13-2 System Characters**

Character	Interpretation
<code>;</code>	Indicates the start of a comment
<code>#</code>	Represents the computer's name
space	Used as a separator
<code>&lt;</code>	Indicates start of a character entered by alias or decimal value
<code>@</code>	Causes system to abandon the line you are typing

Table 13-3 shows the aliases and decimal values you must use if you need to use these characters as control codes.

**Table 13-3 Alias and Decimal Values of System Characters**

Character	Alias	Decimal value
<i>i</i>	<sc>	<59>
#		<35>
space	<sp>	<32>
<	<<	<60>
@	\@	<64>

Enter a greater than sign (>), and then enter the at (@) symbol by typing it, but only if you first type a backslash (\) character. This causes the console to treat the @ as a normal character, rather than the command to abandon the current line, and put it in the profile you are editing.

## Converting Text with Accents and Diacritical Marks

Wire services that deliver 8-bit codes may send only a two-character sequence, typically a nonspacing diacritical mark and a character. Avstar NRCS wire programs do not recognize these two character sequences. However, each nonspacing diacritical mark can be mapped to one of the recognized accent-mark/enable-accent codes listed in Table 13-4.

The bits selection defines a mapping table to convert received wire codes into codes used by the system. The system applies ASCII conversions to codes 0-127 and the DEC Multinational Character Set (MCS) conversions for codes 128-255. When you specify *ibm*, a selected map is used to convert an IBM code into a DEC MCS code.

To send accented characters, wire services send a sequence of three characters, typically an accent mark, an *enable-accent* code, and a

character, when the accented character is not included in the wire service character set.

Avstar NRCS wire programs convert these accent sequences into accented characters in the DEC MCS. When the wire profile contains an `accent` definition (typically a backspace character), the wire program provides the following conversions.

**Table 13-4 Wire Program Conversions**

Mark	Accent displayed	Supported characters
"	Umlaut (¨)	Ä ä Ë ë Ì ì Ö ö Û ü Ÿ ŷ
`	Grave (´)	À à È è Ì ì Ò ò Ù ù
´	Acute (´)	Á á Ê é Í í Ó ó Ú ú
^	Circumflex (ˆ)	Â â Ê ê Î î Ô ô Û ú
~	Tilde (˜)	Ã ã Ñ ñ Õ õ
,	Cedilla (,)	Ç ç
*	Ring (°)	Å å
/	Slash (/)	Ø ø

If the wire service uses ISO characters, your map lines should look like these:

```

; ISO accent mapping
map <193> `<backspace> ; grave
map <194> '<backspace> ; acute
map <195> ^<backspace> ; circumflex
map <196> ~<backspace> ; tilde
map <200> "<backspace> ; umlaut
map <202> *<backspace> ; ring
map <203> ,<backspace> ; cedilla
map <233> <216> ; slash-o
map <249> <248> ; slash-o

```

The ISO character set includes specific values for the Ø and ø characters, rather than building them out of two other characters. These characters are mapped to corresponding values in the DEC Multinational character set (<216> and <248>).

Similar lines using different values are used for ANPA characters:

```

; ANPA accent mapping
map <196> '<backspace> ; acute
map <197> ,<backspace> ; cedilla
map <198> ^<backspace> ; circumflex
map <199> `<backspace> ; grave
map <200> /<backspace> ; slash
map <201> ~<backspace> ; tilde
map <202> "<backspace> ; umlaut
map <203> *<backspace> ; ring
map <204> _ ; underline

```

The accent sequence that results from this mapping is paired with the character that follows it in the wire service transmission and checked to see whether the specified accent is available with that character. For instance, the ISO sequence <202>a is mapped to \*<bs>a by the Avstar system, which matches the â character.

The ISO sequence <202>u is displayed as \*u, because the ° accent is not available for use with the letter u. The sequence <202>u received from an ANPA wire is translated to <bs>u by Avstar and displayed as ü in the story.

## Distributing Stories from the Wire

Wire programs route each wire story they receive to an appropriate queue, based on which wire service sent the story and the story's distribution code. This distribution system ensures that all sports-related wire stories are sent to the sports wire queue, weather forecasts are sent to the weather wire queue, and so on.

This section explains the method wire programs use to categorize and distribute wire stories, and provides a process for customizing the distribution of wire stories to keep up with the changing needs of your newsroom.

## Defining Distribution of Wire Stories

Each wire program must have its own entry in the wire distribution story. The wire distribution story is the first story in the `SYSTEM.WIRES.DISTRIBUTION` queue. If there is more than one story in that queue, the first story is used. This entry consists of up to 150 instructions that tell the wire program how to distribute wire stories, based on their wire service codes.

Each line in the story is called a *distribution line*. Each distribution line consists of a distribution name, a destination queue, and an optional notification priority constraint.

Here is a sample wires distribution story:

```

AP#####a#   wires.us-national
AP#####b#   wires.us-national
AP#####d#   wires.features
AP#####e#   wires.briefs
AP#####f#   wires.features
AP#####h#   wires.business
AP#####h#   wires.briefs
AP#####i#   wires.us-national
AP#####l#   wires.features
AP#####m#   wires.business
AP#####o#   wires.weather
AP#####p#   wires.us-national
AP#####q#   wires.sports-score
AP#####r#   wires.rundowns
AP#####s#   wires.sports-stories
AP#####t#   wires.rundowns
AP#####v#   wires.advisory.other
AP#####w#   wires.washington
AP#####    wires.all
AP#####    wires.unknown

```

## Creating a Distribution Name

Each distribution line must begin with a distribution name that sets the criteria a story must meet to be sent to the destination queue specified in that instruction. Begin the distribution name with the wire program's source name, as specified in the configuration file. Otherwise, the wire program ignores that instruction.

The source name is the contents of the fifth column for the wire in the `/site/config` file. The following wire on port 18 is named AX:

```
wire 18 9600-8n anpa7 AX - ;ap express
```

Follow the source name with the wire service code you want that instruction to handle. There must be no spaces between the source name and code. Within the code, specify the category, service level, selector, and priority codes a story must have for the wire program to send it to the destination queue specified in the instruction.

In most cases, you need only specify the category code to distribute wires. You generally use only the source name and category code posi-

tions in the distribution name and fill the remainder of positions in the distribution name with pound sign (#) characters, which the wire program ignores.

For instance, to create a distribution name that matches all Associated Press sports stories you receive, begin the distribution name with the source name of the wire program that handles the AP wire (AP in this example) and put `s` in the ninth (second-to-last) position. It does not matter what is in the remainder of the distribution name, so fill those positions with # characters: `AP#####s#`.



*When you enter a distribution name, type the wire program's source name in the same case as it is defined in the configuration file. Similarly, wire category and priority codes are always sent in lowercase, so enter them that way in your distribution names.*

## Identifying a Destination Queue

Follow the distribution name with the full pathname of the destination queue where you want the wire program to put any stories that match the distribution name. For instance, to have the wire program send every story that matches `AP#####s#` to `WIRES.SPORTS`, specify that queue:

```
AP#####s#           WIRES.SPORTS
```

If you create a distribution line with a destination queue that does not exist, the system creates the queue when you save the distribution story. A queue created this way inherits directory traits assigned to the Wires directory, including the purge interval. If you want a different purge interval, use the Directory Properties dialog box to change it. See “Changing Database Traits” on page 5-21, “Directory/Queue Properties Dialog Box” on page 5-25, and “Choosing Queues to be Purged” on page 5-40 for more information.

## Changing Notification Priority

Normally, when your system receives a story the wire service has marked as either `BULLETIN` or `URGENT`, it notifies everyone logged in. You can change the priority level a story must have to merit notifi-

cation. Set an optional notification priority constraint in the distribution line.

You can set four notification priority constraint levels in each distribution line. Each level restricts user notification to a higher degree. By convention, type them in uppercase.

**Table 13-5 Priority Constraint Levels**

Priority Level	Description
URGENT	The lowest constraint level. Any story marked <code>flash</code> , <code>bulletin</code> , or <code>urgent</code> by the wire service causes users to be notified if this constraint level is specified.
BULLETIN	The next-lowest constraint level. Any story marked <code>flash</code> or <code>bulletin</code> causes users to be notified if this constraint level is specified.  BULLETIN is the default priority level; if you do not specify a notification priority constraint in a distribution line, any story that matches the distribution name in that line and has this level or higher priority causes notification.
FLASH	The second-highest constraint level. When you set a distribution line to this constraint level, stories that match the distribution line's distribution name must have a priority of "flash" in your system to notify users.
SILENT	The highest constraint level. It prevents your system from notifying users of the arrival of any story processed by that distribution line. No story ever has a high enough priority to override this constraint level.

## Setting the Transmit or Always Options

You can apply two other options, `TRANSMIT` and `ALWAYS`, to a distribution line.

### TRANSMIT

This option is used on distribution lines that send stories to queues scanned by distribution servers. It causes the wire pro-



gram to give the story a distribution code that matches its wire service code before sending it to the destination queue. This way, a distribution server that scans the destination queue can distribute the story, based on its original wire-service code.

#### ALWAYS

If you want the wire program to distribute a copy of every story it receives to a certain queue, place `ALWAYS` on the distribution line that sends stories to that queue. The word `ALWAYS` may follow or precede a notification priority constraint. Most often, this is used to send a copy of every story received to `WIRES.ALL`:

```
AP##### wires.all ALWAYS
```

## Adding a Distribution Line

Although your distribution story was set up for you when your system was installed, you may need to modify it by changing an existing distribution line or adding a new distribution line to the story. Distribution lines that send the same category of story to different queues may be scattered throughout the story.

For instance, to have the wire program that handles the AP wire service put a copy of each Associated Press brief it receives in `WIRES.BRIEFS` and notify you whenever it puts an AP brief marked urgent in this queue, add the distribution line shown here:

```
AP#####d# wires.briefs URGENT
```



**When you add a new distribution line, you must add it above any distribution lines for `WIRES.UNKNOWN` or `WIRES.ALL`.**

## Avoiding Hidden Categories

When you add lines to your wire distribution story, you may get hidden category error messages when you save the story. These errors occur when the line or lines you have added are not in the proper posi-

tion relative to other lines in the story. The following example illustrates this problem and shows how to fix it.

Suppose you had this line in your wire distribution story:

```
UP#####g#      wires.regional
```

In addition to this line, you wanted to add a line like this one:

```
UP#xyz##g#      wires.briefs
```

While this may seem straightforward, you get a hidden category error message if you add the new line following the existing one. To understand why, look at the two lines together.

```
UP#####g#      wires.regional
UP#xyz##g#      wires.briefs
```

Suppose a story came from the wire service with a distribution code of `UP0xyz0ogr` or something else that would ordinarily match your new `WIRES.BRIEFS` distribution line. The problem arises because the story also matches the `WIRES.REGIONAL` line above that one, and, since the wire distribution program stops checking the distribution story after it finds a match, it never gets to the `WIRES.BRIEFS` line.

To avoid this problem, ensure that you arrange any distribution lines that have common elements so the more specific line comes first in the story, like this:

```
UP#xyz##g#      wires.briefs
UP#####g#      wires.regional
```

This rule of distribution does not apply to distribution lines that have the same code but distribute the stories to two different queues. These lines still match the appropriate wire stories, as long as they are placed together in the distribution story.

## Using the `WIRES.ALL` Notification Priority

Your system sends only one notification for a story, even if the story matches two or more distribution lines. However, since each line may

have a different notification level, the system chooses the lowest notification level and uses it to determine whether to send a notification.

This is important when considering the distribution line for `WIRES . ALL`. By definition, each story received is distributed to this line as well as others. Consequently, the notification priority set for this line can also affect how notification levels are interpreted for other lines.

For instance, if the distribution line for `WIRES . ALL` has a notification level of `BULLETIN`, any other distribution line set to a notification level of `FLASH` or `SILENT` (higher levels) is downgraded to `BULLETIN`.

However, by setting the distribution line for `WIRES . ALL` to `SILENT` (the highest notification level), you can ensure that all other distribution lines for that wire have notification levels lower than `WIRES . ALL`. This ensures their notification levels take precedence over those for `WIRES . ALL`.

## Distributing Unknown Wires

A wire program puts stories that do not match any of its distribution lines in the queue `WIRES . UNKNOWN`. Most systems have a distribution line that uses a wildcard distribution code and the `ALWAYS` option to send a copy of every story to `WIRES . ALL`. As a result, no story is unknown as far as the wire program is concerned.

In many cases, you want to send stories that do not match normal distribution codes to both `WIRES . ALL` and `WIRES . UNKNOWN`. To do this, you need a special distribution line that sends a copy of any wire that does not match a normal distribution code to `WIRES . UNKNOWN` before sending it to `WIRES . ALL`.

If your system uses more than one wire service, the distribution lines for each service should be in separate groups. Each group should have its own distribution line for `WIRES . UNKNOWN` at the bottom of that group of distribution lines, just above that group's distribution line for `WIRES . ALL`.

## Maximum Number of Lines

Each wire program's entry in the distribution story can contain up to 150 distribution lines. To fix a distribution story so your system can use it, divert distribution wires to the distribution server so the total for each wire program is no more than 150. Once you remove excess distribution lines and save the story, your system should accept the distribution story and begin using it.

## Mailboxes

Your system was installed with the reserved `distribution` mailbox assigned to the queue `SYSTEM.WIRES.DISTRIBUTION`. This mailbox allows the wire distribution checker (a server program) know when you make changes to the distribution story. The checker in turn notifies the wire programs that the distribution story has changed. This is the only way the wire program knows to incorporate changes you make. See "Using Mailboxes" on page 14-14 for more information.



*Do not assign a different mailbox to this queue or assign this mailbox to any other queue.*

If you want everyone logged in on the system to be notified when a wire story is placed in `WIRES.ADVISORY.PRIORITY`, you must assign the reserved `all` mailbox to that queue. Mailboxes are assigned to queues using the Queue Properties dialog box. See "Changing Database Traits" on page 5-21 and "Reserved Mailboxes" on page 14-16 for more information.

## Purge Intervals

Nowhere in your database are proper purge intervals as important as in the Wires directory. All queues in the Wires directory should have roughly similar purge intervals. If not, the system may not reclaim space used by old wire stories efficiently.

Never use the wire distribution story to distribute wire stories to queues not purged. Doing so can eventually lead to a low-on-space condition as the queues fill up with old stories.

## Internationalization

You can modify the `FLASH`, `URGENT`, `BULLETIN`, `SILENT`, `ALWAYS`, and `TRANSMIT` options by changing appropriate entries in the words dictionary (`/site/dict/words`).

You must use either the `makeccutab` or `maketab` command for the system to recognize dictionary changes.

The system expects each option to begin with a unique letter and looks at the first letter of the word to determine what it is. The system would find `TRANSMIT` to be the same as `TX`.

## Operating Wire Keyword Searches

In addition to receiving and distributing wire stories, a wire program can also search stories it processes for keywords you want. When it finds a story that satisfies one of your search rules, it copies that story to a queue you specify. Using wire keyword searching can be an easy way to collect stories of interest to specific writers in your newsroom.

## Setting up Keyword Searching

Utility (server) programs, called keyword checkers, can do keyword searching in the same way as wires. They are listed differently in the search job list, but otherwise searching and story distribution is handled the same way.

See “Adding a Keyword Server” on page 14-54 for more information.

To set up a keyword search for a wire or keyword server, do the following:

1. Check the search job list for existing entries.

The search job list is the first story in the `SYSTEM.WIRES.KEYWORDS` queue.

Each job list entry uses the following format:

```
*[<number of queues>] [<name>] [< name>]
[<queue name>]
```

...

```
[<queue name>]
```

Begin each entry with an asterisk (\*). Include a number, which indicates the number of queue names that follow. Type `5` here if the entry is for a wire program—this is also the default value.

Follow the asterisk and number combination with the device name of the program that uses this entry or a source name for a wire. List two or more names if you want more than one program to use the entry. Names are specified in the configuration file.

Follow the first line with one or more lines that specify queues containing keyword stories you want the program to use. A keyword server can have as many as 20 queues.

If the program that uses the entry is a wire program, you can list as many as five different queues here.

## 2. Create a keyword queue.

By convention, keyword stories are placed in queues in `SYSTEM.WIRES`, so create a queue called `SYSTEM.WIRES.APKEYWORDS`.

See “Adding a Directory or Queue” on page 5-3 for more information.

## 3. Assign the reserved `keyword` mailbox to the new keyword queue.

If this mailbox is not assigned to this queue, your system cannot notify the keyword checker (one of the utility programs called servers) when you add new search rules to the keyword story or modify existing ones. The keyword checker will not examine your changes for mistakes or notify programs that use that keyword story of changes you made.

See “The Keyword Mailbox” on page 13-48, “Changing Database Traits” on page 5-21 and “Reserved Mailboxes” on page 14-16 for more information.

4. Add an entry to the search job list.

Ensure the keyword story queue you just created is listed in the wire program's search job list.

Avstar NRCS looks for the keyword story in the `SYSTEM.WIRES.APKEYWORDS` queue, so the wire program's search job list should look similar to this:

```
* AP
system.wires.apkeywords
;
* kwdserver1
system.keywords.shuttle
system.keywords.recycle
```

When you save the story, the keyword checker examines your changes for errors. Then it notifies all wire programs that keyword changes were made.

The number of queues you specify determines how many unique keywords can be in each story. The formula for this is the maximum number of unique keywords a wire program can use (250) divided by the number of stories you declare. For instance, listing two queues would mean each story could have a maximum of 125 unique keywords.

See “Additional Information about Search Jobs” on page 13-42 for more information.

5. Create a keyword story.

Wire and keyword server programs assume the first story in the queue is the keyword story. If there is more than one story, ensure that your keyword story is at the top of the queue.

6. Add search rules to the keyword story. See “Keyword Searching” on page 13-43 for more information. Save the keyword story.

The keyword checker examines the story for errors and creates any destination queues that do not exist. If the keyword checker finds an error, it sends a message to your workstation.

The most important of these messages advises you the story contains a destination queue that does not have a purge interval.

Using a destination queue that is not purged does not prevent the keyword story from working, but you risk having any program that uses the rule set collect stories in that queue until the system runs out of space. We strongly recommend that all your destination queues have purge intervals.

The messages you see after saving a keyword story are advisory messages and do not prevent the keyword story from being used. Once you have viewed all the messages, the program for which the keyword story was designed begins using the rule sets in that keyword story to search the stories it processes, as long as an entry for that program has been made in the search job list.

## **Additional Information about Search Jobs**

The following section provides additional information about search job list features and important considerations.

### **Suppressing a Search**

To prevent a program from doing any searching, create an entry for the program that consists of an asterisk only and the program's device or source name, or do not include an entry for the program in the search job list. If your search job list contains a default entry, you need to include entries for each wire and keyword server program you want to keep from doing any searching.

### **Default Entry**

A default entry is one that begins with an asterisk, but does not specify a device or source name. Often, a search job list contains a default



entry located at the bottom of the story. If a program is unable to find an entry that contains its device name, it uses the default entry.

Your system uses a special utility (server) program, called keyword checker, to check any changes you make to the search job list. For the system to notify the keyword checker when you modify the search job list, assign the reserved system mailbox `keyword` to the `SYSTEM.WIRES.KEYWORDS` queue. See “Changing Database Traits” on page 5-21 and the “Mailbox section” on page 5-39 for more information.

## Keyword Limitations

Each wire program can have search rules that contain no more than 250 unique keywords. Divide the search rules among as many as five keyword stories in different queues. The maximum number of unique keywords that you can use in each queue’s keyword story is 250 divided by the number of queues you specify in the program’s search job-list entry.

In a similar way, keyword servers can use search rules that contain a total of 1000 unique keywords. Divide these search rules among as many as 20 different queues. The number of unique keywords allowed in each keyword story is equal to the total number allowed for that keyword server divided by the number of queues you include in the keyword server’s search job list entry.

## Keyword Searching

Each keyword story is a collection of rule sets. Each rule set contains the search rules you want your program to use to identify stories relating to a particular topic.

The format each rule set must have is as follows:

```
destination <queue name>
<search rule>
[<search rule>]
...
[<search rule>]
```

Each rule set begins with a destination line that names the queue where copies of stories that satisfy any of the rule set's search rules are placed. This line must begin with the word `destination`, followed by the full pathname of the destination queue. Two rule sets cannot use the same destination queue.

Follow the destination line with one or more lines that contain search rules you want a particular wire and keyword server program to use to search. Each search rule occupies a separate line in the rule set.

For instance, the rule set in the following example begins with a destination line that tells the program to put copies of stories that satisfy either of the rule set's search rules into `WIRES.KEYWORDS.DRUGS`. The two lines that follow contain the search rules the program uses to find stories relating to drugs:

```
destination      wires.keywords.drugs
    panama AND drugs
    drugs NOT pharmacy OR <drugs AND smuggling>
```



**Never create a destination line that sends stories to a personal queue or any queue not purged. Having a program send stories to such queues can lead to an out-of-space condition.**

Each search rule consists of one or more keywords you want a wire or keyword server program to use to locate stories on a particular topic.

Search rules can be up to one line in length and contain as many keywords as fit on that line. You can have more than one search rule defined for each destination queue. The only limitation is that

you cannot exceed the number of unique keywords allowed for the story. For details on the maximum number of unique keywords, see “Keyword Limitations” on page 13-43.

If your search rule contains more than one keyword:

- Include a space before and after each keyword so the system can identify the start and end of each keyword. You can change default delimiters by customizing the dictionary translation for the `W_DELIMITERS` token in the words dictionary. Use the `makecutab` command or the `maketab` command to allow the system to recognize dictionary changes.

Delimiters defined by `W_DELIMITERS` are used by the wire, seek, and keyword server programs to find the start and end of words in stories. These isolated words are matched with keywords from the search list. Definition of the delimiters is not related to the task of constructing search rules.

- Include the AND, OR, or NOT operator between two keywords:

AND requires that a story contain multiple keywords. For instance, a rule like “drugs AND smuggling” finds only stories that contains *both* of these words.

OR requires that a story have any of multiple keywords. For instance, a rule like “cnn OR panama” finds any story that contains *either* of these words.

NOT specifies that a story must not contain the keyword following the operator. For instance, a rule like “drugs NOT pharmacy” finds any story that has the word “drugs” in it but does not contain the word “pharmacy.”

## Using Parentheses in Searches

Wire and keyword server programs evaluate a search rule from left to right. For instance, to find stories that contain either “air quality,” “clean air,” or “dirty air,” create the following long search rule:

```
air AND quality OR air AND dirty OR air AND clean
```

Because these programs evaluate rules from left to right, they first see if a story contains both “air” and “quality.” If a story does not contain both of these keywords, the program checks whether it contains both “air” and “dirty.” If the story does not, the program checks whether the story contains both “air” and “clean.”

Using parentheses, you can control how a program evaluates the rule. For instance, by rearranging the rule, we greatly simplify it:

```
air AND (quality OR dirty OR clean)
```

You can use as many sets of parentheses as necessary to have the program evaluate the search rule.

For instance, to create a search rule that finds any story that contains either the word “smog” or the words “air” and “quality,” but not “pollen” or “airline,” build a rule like the following:

```
smog OR ((air AND quality) NOT (pollen OR airline))
```

After you add your search rules to the keyword story and save, the keyword checker examines your changes for errors. If there are no errors, it notifies any program that uses that keyword story of what you have done so the program can use your changes.

## Tips on Building Search Rules

- Each keyword must be a single word; two words without an operator between them generate an error. For instance, you must use `space AND shuttle` rather than `space shuttle` to find stories about the space shuttle.
- A search rule can be up to one-line long, and each keyword can be up to 10-characters long. Searches are not case-sensitive, so you can use uppercase or lowercase letters.
- Be careful using keywords that form the first part of larger words. For instance, “air” finds stories that contain words like “airline” and “airports” as well as “air.” To use a keyword like “air,” include the NOT operator to exclude other words that begin with that keyword, such as `NOT airline`.

However, wire and keyword server programs do not find keywords when they appear in the middle or at the end of a word. For instance, searching for “ports” does not locate stories that contain words like “airports” or “sports.”

The success of your search depends on how well you design the search rules; they should be specific enough so programs that use them find only stories that relate to your topic.

Scanning stops either at the end of the story or after a maximum of 50 keywords are found. Then the search rules are used on the set of located keywords. The program abandons that rule set when it finds a search rule the story satisfies or when it exhausts all search rules in the rule set.



*You can increase the efficiency of your searches by putting rules that are most likely to find stories at the top of each rule set. Put the least complex rules at the top.*

## User Notification

You can have the system notify you when a program finds a story that contains one of the keywords for which you are searching. Assign a group you are in to the notification group trait of the queue(s) where the program sends the stories. See “Notification Group” on page 6-25 for more information. Your system then notifies you every time the program places a story in one of these queues.

## Removing a Rule Set

When a rule set becomes obsolete, remove it. Because a large number of keyword searches slows your system, removing unused rule sets helps maintain system efficiency.

If you are no longer using the rule set’s destination queue, remove it. This frees up space in the database and helps keep it from being cluttered up with unused queues.

## Sending a Story to More Than One Queue

Wire and keyword server programs check stories they process against each rule set in the keyword stories they use. When the program finds a match, it sends a copy of the story to the rule set's destination queue and continues searching, using the remaining rule sets. Only the first 50 words identified in the story will be used when checking the rule sets.

So, you can have a story copied to more than one destination queue by repeating the same search rule in different rule sets.

## Default Directory Paths

You usually need the full pathname to specify each rule set's destination queue. You must do this for the program performing the search, to locate the destination queue.

Sometimes, all rule sets in a keyword story contain destination queues in the same directory. In this case, set a default directory path for the keyword story and then specify the destination queues by name rather than using the full pathnames.

To specify a default directory, enter the queue's default directory in the search job list entry that uses the keyword story. Follow the queue name that contains the keyword story with the default directory path-name.

Use this feature with group security. You can allow users to manage their own keyword searches but control where they send the stories.

## The Keyword Mailbox

See "Changing Database Traits" on page 5-21 and "Reserved Mailboxes" on page 14-16 for more information.

Assign the reserved `keyword` mailbox to the queue `SYSTEM.WIRES.KEYWORDS`, which contains the search job list. Otherwise, your system cannot notify the keyword checker when you make changes to the search job list.

Always assign the `keyword` mailbox to queues that hold your keyword stories. If this reserved system mailbox is not assigned, your sys-

tem cannot notify the keyword checker when you create new search rules or modify existing ones.

When you modify an existing search rule or create a new one, the keyword checker alerts programs that use that rule about the change. If the queue does not have the `keyword` mailbox assigned to it, the keyword checker cannot do this. Consequently, programs that use the search rule you created or modified will not know about your changes until you restart them.

## Keyword Checker Messages

The following messages from the message dictionary may appear when you modify and save a keyword story:

Could not enter the queue (M\_BADQUEUE)

The specified destination is not accessible or an error occurred while creating or entering the queue.

Default keyword list must be last (M\_WNOTLAST)

Placing the default keyword list anywhere but at the bottom of the story will prevent those keyword lists that follow it from being used.

Duplicate destination (M\_DUPEDEST)

The specified destination has already been used for another keyword list.

Keyword too long (M\_KWDLONG)

Length of the keyword string is longer than maximum length allowed.

Maximum file number bad (M\_FILENUM)

The number of keyword stories has exceeded the maximum allowed.

missing (M\_MISSING)

The keyword story does not understand the specified rule set. Either an operator is missing (AND, OR, or NOT) or an unmatched parenthesis exists.

No destination found (M\_NODEST)

No destination is specified for a given rule set.

Not a queue (M\_NOTQUEUE)

The specified destination is not a valid queue name. Check whether it is misspelled, or whether it is a directory.

Queue is never purged (M\_PURGEZERO)

The specified destination queue is never purged, which means stories may accumulate in it and eventually cause a space problem in the database.

Syntax error (M\_SYNEROR)

The keyword checker does not understand the specified line. Check whether format of the line is correct.

System error (M\_SYSERROR)

A system error has occurred that caused the keyword checker to terminate.

Too many keyword distribution files (M\_KWFMAX)

The number of keyword distribution files exceeded the maximum allowed.

Too many keywords (M\_KWDMAX)

The total number of unique keywords processed by the keyword checker has exceeded the maximum allowed.

Unexpected (M\_UNEXPCTD)

The specified line has extra punctuation, such as an unmatched parenthesis. Ensure that parentheses appear only in pairs.



# CHAPTER 14

## *Servers*

Do not confuse servers in this chapter with server hardware or file servers, which are computers with the Avstar database and running the Avstar application software. Those computers are referred to in this chapter as Avstar Servers, server A, server B, and so forth. This chapter explains setup and use of utility programs, also known as *servers*. There are various types of these server programs, which perform a multitude of tasks. This chapter contains the following information:

- Overview
- Adding a Server Program to the System
- Job Lists: Queues, Stories, and Commands
- Action Servers
- Distribution Servers
- Parallel Wire Servers
- Keyword Servers
- System Servers
  - Seek Servers & Fast Text Search (FTS) Servers
  - Mail Servers
  - Print Servers
- Networking Two or More Servers Using Rx/Tx Links

## Overview

*Servers* are utility programs, which automatically perform various actions on stories in the Avstar database. Some are general purpose and do jobs you define; others are narrowly defined, needing no instruction from you. You can connect (network) server programs in separate systems together using Rx/Tx links.

These various types of server programs are explained in more detail in individual sections later in this chapter.

The types of servers available to you are:

- **Action servers** - general-purpose servers that move, duplicate, remove, or replace stories added to or edited in certain queues
- **Distribution servers** - distribute stories to different queues, based on each story's distribution code
- **Parallel wire servers** - allow you to have a wire distribution backup in case you have problems with the usual wire port
- **Keyword servers** - perform keyword searches on stories added to or modified in the designated queues and then copies or moves results to a queue that you specify
- **System servers** include:
  - **Seek servers** - perform searches
  - **Fast Text Search (FTS) servers** - perform fast text searches
  - **Print servers** - format stories sent to printers
  - **Mail servers** - handle mail processing

Although listed as a special device in the configuration file, txnets are very similar to servers in function and configuration. See "Networking Two or More Servers Using Rx/Tx Links" on page 14-88 for more information.

## Adding a Server Program to the System

The overall process of creating a server is the same for all server utility programs.

Guidelines for adding a server to Avstar NRCS are:

1. Check the configuration file (`/site/config`) and choose the next available device number for the server program, from the range of 3-digit numbers reserved for use by your system's server programs, such as 256 to 300.
2. Add the server program to the `/site/config` file on each Avstar Server—such as server A and server B—in your system. Changing the configuration file requires the use of `ed`, the UNIX line editor. See “`ed`, the Line Editor” on page 10-1 and “Changing the Configuration File” on page 11-15 for more information.
  - a. Select *all* Avstar Servers. See “Selecting Servers” on page 2-5 for more information.

- b. Use the `ed` command to open and edit the configuration file, by typing:

```
ed /site/config
```

Press Enter. A message similar to the following appears:

```
editing /site/config
1259
```

- c. Add the server program's device number to the `servers` line in the configuration file. For instance:

```
servers 257 258 259 260
```

The device number 260 is added to the `servers` line in this example.

- d. Add a configuration line for the server program in the host definition belonging to the Avstar Server that will run the server program. This line should contain the mailbox number assigned to the server program, when necessary.



*Divide your server programs evenly among your Avstar Servers to distribute the load they put on your system. Additionally, ensure that you also add the configuration line for the server program to alternate host definitions for your Avstar Servers. This ensures the server program can run on the surviving computer should one of your Avstar Servers stop functioning. See “The Site Configuration File (/site/config)” on page 11-7 for more information.*

The format for server programs' configuration lines are similar:

```
server <device#> <type> <mailbox> <device name>
```

Parameter	Description
device #	The device number assigned to the server program. This 3-digit number must also be listed in the <code>servers</code> line in a host definition.
type	The type of server program, such as <code>action</code> , <code>distribution</code> , <code>parallel</code> , <code>keyword</code> , <code>seek</code> , and so forth.
mailbox	The mailbox the server program uses. Valid standard mailbox numbers are 1 through 4096. This number typically matches the server program's device number. See “Using Mailboxes” on page 14-14 for more information.
device name	To give the action server a device name, enter that name here (up to 8 characters). If not, enter a hyphen.

Comments appearing after the semicolons (;) are optional.

The following are sample configuration lines for various server programs:

```
server 256 action 256 actphon ;action svr
server 257 distribution 257 devname1 ;dist server
server 258 parallel 258 devname2
server 259 keyword 259 key1 ;keyword server
server 260 seek 260 seek ;seek server
server 261 ftsseek 261 - ;fts searches
server 262 ftsindex 262 - ;fts indexing
server 263 print 263 - ;print server
```

Do *not* use an uppercase (W) in step 2e. See “ed, the Line Editor” on page 10-1 for more information.

- e. When you finish making changes to the configuration file, save your changes and exit the UNIX line editor by typing:

```
w
1279
q
```

Information pertaining to steps 3, 4, 5, and 6 vary depending on the type of server program. These steps are described in more detail in individual sections for each server program later in this chapter.

3. Create a job list queue in the System directory of the Avstar database. See “Adding a Directory or Queue” on page 5-3 for more information. The name and location of the job list queue in the System directory varies, depending on the type of server program.
4. Create a job list story with defined tasks. See “Job Lists: Queues, Stories, and Commands” on page 14-7 for more information.



*Check your job list before you save the story. When you start the server, it checks for syntax errors in time intervals, but does not catch errors such as misspelled queue names. You must make sure that everything is correct.*

5. (Optional) Additional queues and stories may be required for some utility programs, such as distribution and parallel servers. See individual sections for each server program later in this chapter for more information.

6. Assign mailboxes to queues, if necessary. The queue's mailbox number should match the server program's mailbox number, which appears in its configuration line in the `/site/config` file.
7. (Optional) Test your configuration changes. See "Testing the Site Configuration File After Changing" on page 11-9 for more information.
8. Reconfigure the system. You do not need to stop anything to reconfigure the system.
  - a. Select the master computer (typically server A). See "Selecting Servers" on page 2-5 for more information.
  - b. Become a superuser. See "Becoming a Console Superuser" on page 3-2 for more information.
  - c. Type:  
`AVSTAR-A# offline`
  - d. Type:  
`AVSTAR-A# configure`
  - e. When the prompt returns, bring the system online again by typing:  
`AVSTAR-A# online`
  - f. Exit from superuser mode. (CTRL-D)

A message similar to the following will appear:

```
A Wed Oct 4 00:18:58 2000 msg System being configured
```

9. After you see the `System being configured` message, select the Avstar Server on which you configured the server program. See "Selecting Servers" on page 2-5 for more information.
10. Start the server program using the `restart` command in the following format: `restart <device#>`. For instance, to restart the server program with device number 257, type:  
`AVSTAR-B: restart 257`

11. (Optional) Back up site files. See “Backing up Site Files” on page 20-25 for more information.

## Job Lists: Queues, Stories, and Commands

Server programs, such as action and distribution servers, require a *job list queue* to hold its *job list story*, a set of instructions telling the server program what you want it to do. Job lists for each action, distribution, parallel, and keyword server must be in its own job list queue in the `SYSTEM.CONFIGURE` directory. The job list story must be the first story in the appropriate queue. Also, the job list queue name must match the server’s device number.

For instance, an action server with device number 257 would have a job list queue named 257 in the Configure folder of the System directory. The full pathname would be: `SYSTEM.CONFIGURE.257`. You may want to add a description of the server’s function as part of its queue’s name. Do this by using a hyphen, such as `SYSTEM.CONFIGURE.257-DUP-TO-ARCHIVE`. Further examples are provided in procedures for adding various server programs in their corresponding sections later in this chapter.

## Defining Tasks for Servers

The job list story outlines *tasks* the server program is instructed to execute. There are two types of tasks: *mailbox tasks* or *timed-interval tasks*. Most servers execute mailbox tasks, while timed-interval tasks are executed by action servers and tx links.

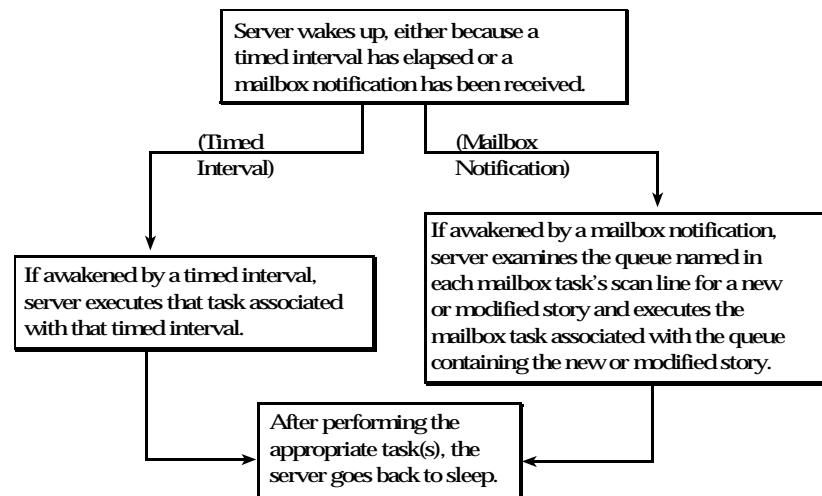
Both tasks must have a *scan line* that identifies the queue with which the task is associated. See “Adding a Scan Line in a Job List Story” on page 14-9 for more information. Follow this scan line with job list commands you want the server to execute if stories have been added to or modified in the task’s scan queue. These commands are the task’s *com-*

*mand set*. See “Defining a Server Command Set” on page 14-10 for more information.

Mailbox tasks are activities the server program is prompted to do whenever something happens to a queue or its contents. Timed-interval tasks are activities an action server or tx link is instructed to do at certain times.

The server program is inactive and said to be “asleep” until it is “awakened” to execute a job list, either when scheduled to (a timed-interval task) or when stories are added to or changed in a queue (mailbox task).

The sequence of actions followed by an action server is shown in Figure 14-1.



**Figure 14-1 Sample Flow of Events for an Action Server**

The difference between the two tasks is that a timed-interval task begins at specified time intervals and a mailbox task does not. Also,



timed-interval tasks are only associated with action servers and tx links, while mailbox tasks are used by other utility programs.

## Adding a Scan Line in a Job List Story

Both mailbox and timed-interval tasks must have a scan line that identifies the queue you want associated with the task. The server performs commands in the task's command set on any new or modified stories in this queue. A scan line must begin with the word `scan` followed by the full pathname of a queue you want the task to watch. The format is:

```
scan <queue name> [all | priority | everyentry]
```

In a mailbox task, the scan line is the first line in the task. See “Defining Mailbox Tasks” on page 14-13 for more information. In a timed-interval task, the scan line follows the time interval commands. See “Defining Timed-Interval Tasks” on page 14-18 for more information.



*A normal scan line reads stories in its scan queue in chronological order, with the oldest first. To reverse this order, use `bscan` instead of `scan`. Apart from this distinction, `scan` and `bscan` operate identically. Use `bscan` for long queues.*

If more than one queue is being scanned by a server, the scan line only processes 10 stories at a time from each queue before going on to the next one, unless the queue is designated as a priority queue or the `all` option is used. The `all` option overrides the 10 entry limit and will cause all entries in the queue to be processed before the server moves on to the next scan set. The `all` option differs from the priority queue because the scan queue does not preempt other scan sets and there is no limit to the number of scan sets with the `all` option specified.

## Defining a Priority Queue

You can set up a scan queue as a priority queue for processing new stories. This means that the 10-story limit for `scan` does not apply to that queue; all new or modified entries in the queue are processed

before the server moves on to the next queue. If the priority queue has no stories to process, the system will scan each of the queues in normal order, returning to the priority queue between each queue to check for new stories.

To designate a scan queue as a priority queue, add `priority` to the scan line after the queue's pathname. For instance:

```
scan SHOW.SCRIPTS.AM      priority
```

### Defining an Every Entry Queue

The `everyentry` option forces the server to process each entry in a queue, not just modified entries.

To designate a scan queue as an every entry queue, add `everyentry` to the scan line after the queue's pathname. For instance:

```
scan SHOW.SCRIPTS.AM      everyentry
```

## Defining a Server Command Set

Follow a task's scan line with a *command set* containing the commands that you want the server to perform on new or modified stories in the queue associated with that task.

A complete list of these commands are provided in Appendix A; see "Job List Commands" on page A-42. Also, see "Ordered Queues and the Order Command" on page 14-13.

Many servers have unique job list commands that only they can use. However, almost every server can use the following five commands.

```
dup <destination queue name> [distribution code]
```

Copies stories in the scan queue to a queue you specify, optionally including a distribution code. It must be followed by the name of a queue into which you want the server program to duplicate stories.



When a server, such as an action server, duplicates a story, it replaces any old version of the story in the destination queue with the updated version only when the update database trait is applied to the destination queue. Server programs will not issue notifications when *dup*, *move*, or *replace* commands are used on a queue with the update database trait and a notification group.

```
replace <destination queue name>
```

Updates a story in the destination queue only if the original is changed in the scan queue; it does not duplicate new stories from the scan queue into the destination queue. The original story must first be manually duplicated to the destination queue. Also, the destination queue must have the update database trait.

```
move <destination queue name> [distribution code]
```

Moves stories from the scan queue to a queue you specify, optionally adding a distribution code to them. It must be followed by the name of a queue into which you want the server to move stories and be the last instruction in a job list task.

```
remove
```

Deletes the story from a scan queue. It does not send stories to the Dead queue but deletes them immediately. It must be the last instruction in a job list task.



Do not include both the *remove* and *move* command in a single command set. The server program cannot do anything else to the story once it executes either one of these commands. Only one of these commands can appear in each command set, and it must be at the end of the command set.

Here is an example of a command set that has an action server duplicate new or modified stories in `SHOW .SCRIPTS .AM` to `BACKUP .MORNING-SCRIPTS` and then remove them from `SHOW .SCRIPTS .AM`:

```
at 10:00
on mon tue wed thu fri
  scan show.scripts.am
  dup backup.morning-scripts
  remove
```

`number <field name> <length> <error queue name>`

Places a unique number in a specified field name of each scanned story when the story is moved or duplicated to a new location. The command should include the field name in which the number is placed, the length in digits of the number (from 1 to 5), and the name of an error queue where stories can be placed if the number cannot be inserted in their forms.

Depending on how many digits you specify, the story number can range as high as 65535, at which point it resets and starts over at 1. The number is also reset to 1 each day at midnight. Specifying fewer digits results in the number rolling over at 9, 99, 999, or 9999. The number is reset at startup. The version of the story with the number in it does not replace the original story in the scan queue.

### Processing Deleted Stories

Ordinarily, servers process stories that are deleted from their scan queues. This counts as a modification. To override this function so that the server takes no action when a story is deleted, you can include the `ignore-del` command in the job list, preceding any scan lines that you want to ignore deleted stories.

To switch back to the default handling of deleted stories later in the job list, include the `send-del` command at that point.

## Ordered Queues and the Order Command

Action servers and Tx links can order queues by using the `order` command in the job list command set. When `dup`, `move`, or `replace` commands are specified while the order option is enforced, the destination queue will be ordered to mirror the scan queue. The `order` command can be turned on or off. The format is:

```
order [yes | no]
```

The `yes` option turns on the order command and the `no` option turns it off. The `order` command is reset (turned off) at the start of any new scan set within the job list story.



*The `order` command cannot be placed between `open` and `put` commands, which are used by Tx links. Otherwise, it can appear anywhere within a scan set to control the ordering of queues. Both `open` and `put` commands can appear between `order yes` and `order no` commands*

Here is an example of the `order` command used in an action server's job list story:

```
scan  main.rundown.6pm all           ;all option on
      order                          ;order turned on
      dup slave1.rundown.6pm        ;will be ordered
      order no                       ;order turned off
      put archive.rundown.6pm       ;not ordered
      order yes                      ;order turned on
      dup slave2.rundown.6pm        ;will be ordered

scan  main.rundown.11pm             ;all & order off
```

## Defining Mailbox Tasks

For a server program to respond immediately to changes in a queue, you must set up mailbox tasks, by assigning the same mailbox number to the server and queue you want it to watch. Otherwise, the server will not know when changes are made to the queue.

As long as no stories are added to or changed in that queue, the server is inactive or “asleep.” However, when a story is modified in or added to that queue, the system wakes the server by notifying it through the mailbox. On notification, the server program checks its job list for mailbox tasks. For each mailbox task, it searches each scan queue to find modified stories and then executes the instructions in it and goes back to sleep.

A server places some additional load on your system when it wakes and executes a task. With mailbox tasks, you have no control over when the server wakes up. Use this kind of task only when the server must act as soon as a story is added to or modified in a particular queue.

In cases where the action can wait until a time when system use is light, use timed-interval tasks, following the instructions in “Defining Timed-Interval Tasks” on page 14-18.



*All mailbox tasks must precede any timed-interval tasks in a job list. If a queue identified in a timed-interval task has the server’s mailbox assigned to it, the server is awakened when changes are made to the queue but the queue is not scanned. That queue is scanned only at the designated time. The server scans all mailbox task lists whenever it wakes up.*

## Using Mailboxes

The server program and the queue it watches must share the same mailbox. There are two kinds of mailboxes: reserved and standard.

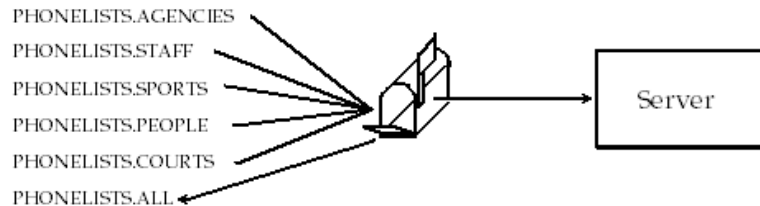
- *Reserved mailboxes* are used with special server programs reserved for system functions. See “Reserved Mailboxes” on page 14-16 for more information.
- *Standard mailboxes* are used by server programs you can configure, such as action servers. There are 4096 standard mailboxes available, numbered 1 through 4096. When you create a server, choose a mailbox number that is the same as the server’s device number.



*These mailboxes are not related to your system's electronic mail feature, but are activation mechanisms or "signal carriers" by which servers are notified to perform predefined tasks.*

When the system detects that a story has been modified in or added to a queue, such as `PHONELISTS.PEOPLE`, it checks to see if a mailbox is assigned to the queue. If so, it notifies any utility program sharing that mailbox that a change has occurred in the queue. That message causes the utility program (in this case, an action server) to perform the task you have assigned to it, such as duplicate information from `PHONELISTS.PEOPLE` to `PHONELISTS.ALL`.

Several queues can be linked to a single server, as shown in Figure 14-2 just by sharing the same mailbox number as the server program.



**Figure 14-2 Linking Queues to an Action Server**

Here's an example of how the mailbox tasks would appear in the job list for the action server in Figure 14-2:

```
scan  phonelists.agencies
      dup phonelists.all
scan  phonelists.staff
      dup phonelists.all
scan  phonelists.sports
      dup phonelists.all
scan  phonelists.people
      dup phonelists.all
scan  phonelists.courts
      dup phonelists.all
```

Mailboxes are not limited to action servers, but can apply to other utility programs, such as distribution, parallel, or keyword servers. For instance, the `WIRES.ALL` queue can be assigned the same mailbox number as a distribution server, or you could set up a `WIRES.AP-SEARCH` queue in Wires with the same mailbox number as the keyword server.

The mailbox is an activation mechanism for a server program, so if a queue has a mailbox number matching a server program, then that server is the one activated or “awakened” whenever something happens to the queue.

### Reserved Mailboxes

Reserved mailboxes are used only with servers reserved for your system to use. Each of these mailboxes links a specific system queue or directory to the server that watches that queue. Each queue that needs a reserved mailbox was assigned the correct one when your system was installed. The information presented here is intended to help you understand how your system uses these reserved mailboxes. Unlike standard mailboxes, reserved mailboxes are referred to by name, not by number. The name and purpose of each of the reserved mailboxes appears in Table 14-1.

**Table 14-1** Reserved Mailboxes

Mailbox	Purpose
All	Assigned to a queue used by your system to notify all users when an event occurs. The All mailbox must be assigned to the queue where urgent wire stories are sent.
Distribution ( <i>dist</i> )	Assigned to the queue containing your system’s distribution story for wires (usually <code>SYSTEM.WIRES.DISTRIBUTION</code> ). This mailbox is used by the server program, known as the distribution checker.



**Table 14-1 Reserved Mailboxes (Continued)**

Mailbox	Purpose
Group ( <i>grp</i> )	Assigned to the system's group queue ( <code>SYSTEM.GROUPS</code> ). This mailbox is used by the server program, known as the group checker, which checks your system's group description stories for errors.
Keyboard ( <i>key</i> )	Assigned to the directory that contains your system's keyboard stories (usually <code>SYSTEM.KEYBOARDS</code> ). This mailbox is used by the server program, known as keyboard checker, which checks your system's keyboard description stories for errors.
Keyword ( <i>kwd</i> )	Assigned to the queue with the search job list (usually <code>SYSTEM.WIRES.KEYWORDS</code> ) or any other queue containing stories with search rules. This mailbox is used by the server program, known as the keyword checker, which checks your system's keyword stories for errors.

## Assigning a Mailbox to a Queue

To set up a particular server, you must assign a standard mailbox for that server to associated queues.

To assign a mailbox to a queue, do the following:

1. Choose a mailbox number (1 through 4096). This number should match the server's device number.
  - a. Use the `list c` console command to ensure no other device is using the mailbox number you have chosen.

For instance, to check mailbox 201, type:

```
list mailbox=201 c
```

Information similar to the following appears:

```
DEV    DEVICE_TYPE COMPUTER  CCU PRINTER SPEED OPTIONS DEVNAME
```

- b. If you see the device configuration header (as shown in step 1a) with no information below it, then no device has that mailbox and you can use that number. However, if configuration information for a device appears below the header, that device has the same mailbox as the one you chose. Therefore, choose another mailbox number and repeat step 1a.



*Multiple queues can share the same mailbox number. To list all queues and directories in the database with a certain mailbox, use this format of the*

*list d command: list mail=<mailbox number> d*

2. Log in to Avstar Workstation as system administrator.
3. Assign the mailbox database trait to the queue.

Mailboxes are assigned to queues in the same way other database traits are—using the Queue Properties dialog box, as explained in Chapter 5. See “Changing Database Traits” on page 5-21, “Database Traits Summary” on page 5-25, and “Mailbox section” on page 5-39 for more information.

## Defining Timed-Interval Tasks

Timed-interval tasks allow you to specify what time you want an action to take place. For instance, you may want an action server to watch a queue but not do anything to the new or modified stories until a time when system use is light. Alternately, you may want an action server to monitor a queue and wait until everyone is done using that queue before performing some action on new or modified stories.

In addition to action servers, timed-interval tasks are also associated with tx links. However, this type of task is not used by other server programs, like distribution and parallel servers. Unlike a mailbox task, a timed-interval task acts like an alarm clock and wakes the server at a time you specify. On waking, the server examines the queues associated with that timed-interval task. If there are any new or modified stories in these queues, the server executes the commands in the tasks.



*All mailbox tasks must precede any timed-interval tasks in a job list. If a queue identified in a timed-interval task has the server's mailbox assigned to it, the server is awakened when changes are made to the queue but the queue is not scanned. That queue is scanned only at the designated time. The server scans all mailbox task lists whenever it wakes up.*

For instance, you may create a timed-interval task that wakes the server at 3 A.M. to look for scripts added to or modified in the `SCRIPTS.TODAY` queue in the last 24 hours. If the server finds any, the task instructs it to move them to `SCRIPTS.HOLDING`.

When you create a timed-interval task, you must begin it with the time interval in which you want it to execute. To create the time interval you want, use one or more of the following job list commands:

`at <hh:mm>`

Sets the time of day when you want the task to wake up the server. Set the time in hours and minutes using a 24-hour clock where 12 A.M. (midnight) is `00:00` and 12 P.M. (noon) is `12:00`. For instance, `at 3:30` wakes the server at 3:30 in the morning. If the time is after midnight but before noon, you do not have to put a 0 in front of the hour.

`every <D.HH>`

`every` followed by a number of days and hours (in `D.HH` format) specifies an interval you want the task to follow when waking the server. For instance, `every 0.12` wakes the server every 12 hours.

`on <day>`

Names the day or days on which you want the task to wake up the server. For instance, `on mon tue` wakes the server on Monday and Tuesday only.

Names of the days of the week are defined in `/site/dict/words`.



*A job list can have a maximum of 32 timed actions. Each day mentioned is considered a timed action. An `at` and/or `every` without an `on` is considered a single-timed action.*

You can combine these commands to create very specific time intervals. For instance, to create a timed-interval task that executes every weekday at 10 A.M., begin the task with the following time interval:

```
at 10:00
ON mon tue wed thu fri
```

### Example of Timed-Interval Tasks

Here is an example of how an action server's job list uses timed-interval tasks to move old scripts out of each show's script queue after the show has aired:

1. Separate timed-interval tasks for each show wakes the server 15 minutes after the show has aired. Because the three shows are not aired on weekends, the timed-interval tasks wake the server only on weekdays.
2. The morning show ends at 8 A.M., so have the first timed-interval task wake the server at 8:15 A.M. and move the scripts in `SHOW.MORNING.SCRIPTS` to `SCRIPTS.TEMP-ARCHIVE`.
3. The noon show ends at 1 P.M., so have the second timed-interval task move the scripts in `SHOW.NOON.SCRIPTS` to `SCRIPTS.TEMP-ARCHIVE` at 1:15 P.M.
4. The evening show ends at 6 P.M., so have the third timed-interval task move the scripts in `SHOW.EVENING.SCRIPTS` to `SCRIPTS.TEMP-ARCHIVE` at 6:15 P.M.

Each timed-interval task must begin with a time interval indicating when you want it to wake the server. This is followed by a scan line identifying the queue you want the server to examine on that day and time. Finally, the scan line is followed by one or more tasks to be performed on stories in that queue at that time.

You can have multiple command sets controlled by a single timed-interval. In this example, the finished tasks look like this:

```
at 8:15
on mon tue wed thu fri
    scan show.morning.scripts
    move scripts.temp-archive
at 13:15
on mon tue wed thu fri
    scan show.noon.scripts
    move scripts.temp-archive
at 18:15
on mon tue wed thu fri
    scan show.evening.scripts
    move scripts.temp-archive
```

## Action Servers

An action server is a general-purpose utility program that automatically performs specified routine tasks, such as move or copy stories added to or edited in certain queues, according to a job list you define. One action server can perform multiple tasks not necessarily related to each other.

- **Example:** You want an action server to update a phone list. You have organized several queues as phone lists: `PHONELISTS.AGENCIES`, `PHONELISTS.COURTS`, `PHONELISTS.PEOPLE`, `PHONELISTS.SPORTS`, and `PHONELISTS.STAFF`. Additionally, you have created another queue as a master phone list, `PHONES.ALL`, that holds all numbers in the individual source phone queues. What you want the action server to do is have `PHONES.ALL` updated whenever anyone adds or changes a number in one of the source phone queues.

- **Example:** You also want the action server to make a daily archive of all the scripts used in your morning, noon, and evening news shows. After each show, you want to move scripts in that show's script queue to the queue `SCRIPTS.ARCHIVE-TEMP`. At the end of the day, you want to move everything in this queue to an archive queue named `SCRIPTS.MAY.01` on May 1, `SCRIPTS.MAY.02` on May 2, and so forth. You want the action server to automatically perform all of these activities for you, rather than having to do them manually.

In the first bulleted example, mailbox tasks are required to achieve the desired results. See Figure 14-2 on page 14-15 for a visual representation of how an action server uses mailbox tasks to maintain a master phone list.

In the second example, timed-interval tasks are needed. See “Example of Timed-Interval Tasks” on page 14-20 for another example of an action server using timed-interval tasks to archive scripts.

## Adding an Action Server

Action servers can perform multiple or single tasks. They can be time-related or respond to a system change. In this section, we will outline a procedure where we will create an action server that performs the following activities:

- Update the master phone list as soon as anyone makes a change to any of the source phone lists
- Move each show's scripts to `SCRIPTS.ARCHIVE-TEMP` at the end of the show

To add an action server to your system, do the following:

1. Choose a device number and mailbox number for the server which will also be used for associated queues. See “Adding a Server Program to the System” on page 14-3 for more information.

2. Add the server (utility program) to the configuration file on each Avstar Server—such as servers A and B—in your system. Details for this step are provided in “Adding a Server Program to the System” on page 14-3.
3. Create a job list queue in the Configure folder of the System directory in the Avstar database. The queue’s name, such as 256, is the number selected in step 1. For instance, the full pathname is `SYSTEM.CONFIGURE.256`.



*Since the action server has a number of tasks, you can end the name with a hyphen and a descriptive word, such as “general.” The pathname would then look like: `SYSTEM.CONFIGURE.256-GENERAL`. See “Creating a New Queue” on page 5-6 for more information.*

4. Create a job list story with tasks for the server.
  - a. Open the job list queue, such as `SYSTEM.CONFIGURE.256`.
  - b. Create a new story, the first in the job list queue, to hold the action server’s job list. See “Creating a New Story” on page 5-9 for more information.
  - c. Add tasks that you want the server to perform; mailbox tasks first, followed by timed-interval tasks. Ensure that each task in the job list story has a scan line identifying a queue to scan, followed by job list commands the server should execute if stories have been added to or modified in the task’s scan queue.

See “Defining Tasks for Servers” on page 14-7 and “Job List Commands” on page A-42 for more information on tasks and job list commands.

Here is a job list story with mailbox tasks:

title	presenter
General Server	
<pre> scan phonelists.agencies dup phonelests.all scan phonelists.courts dup phonelists.all scan phonelists.people dup phonelists.all scan phonelists.sports dup phonelists.all scan phonelists.staff dup phonelists.all </pre>	

Here is a job list story with timed-interval tasks added:

title	presenter	writer
General Server		user5000
<pre> dup phonelists.all At 8:15 On Mon tue wed thu fri scan show.morning.scripts move scripts.temp-archive At 13:15 On Mon tue wed thu fri scan show.noon.scripts move scripts.temp-archive At 18:15 On Mon tue wed thu fri scan show.evening.scripts move scripts.temp-archive </pre>		

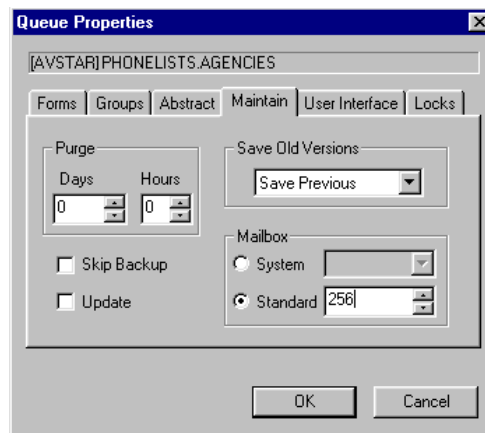


*You can have one server watch many different queues by putting a separate task for each queue in the server's job list. You can have one server perform several tasks or have separate servers perform each task, depending on the task. The only limit to this is that each server's job list cannot be longer than 100 lines.*

*Ideally each queue should have a server, but because of system resource limitations, this organization is rarely possible. Group queues according to the rate they are generally updated (rarely, moderately, or frequently) and assign servers to the groups.*



5. Assign mailbox database traits, using the number matching the job list queue's name and server's device number chosen in step 1, to the appropriate queues. For instance, the job list queue is named 256, so that is the standard mailbox number assigned to queues associated with the job list and action server.
  - a. Open the Queue Properties dialog box for a scan queue, such as PHONELIST.AGENCIES. See "Changing Database Traits" on page 5-21 for more information.
  - b. Click on the Maintain tab. See "Maintain Tab" on page 5-38 for more information.



- c. Select the Standard radio button.
  - d. Type in the mailbox number, such as 256.
  - e. Click OK to save changes.
6. Repeat step 6 as needed for each scan queue in the job list.
7. (Optional) Test your configuration changes. See "Testing the Site Configuration File After Changing" on page 11-9 for more information.
8. Reconfigure the system.

Details for steps 8-10 are provided in "Adding a Server Program to the System" on page 14-3.

9. Start the action server. For instance, to start an action server with device number 256, type:  

```
restart 256
```
10. (Optional) Back up site files with the `sitedump` command.

## Assigning Field Validation

See “Adding Rx/Tx Links” on page 14-93 for more information.

Validation is an option for use with action servers and Tx links. With it, you can provide an action server or Tx link with a list of acceptable values for certain fields. When the action server or Tx link processes a story, it checks the contents of these fields to make sure each contains an entry you have defined as acceptable for that field.

If all fields that you want checked contain acceptable entries, the server or Tx link continues to process the story according to the command set for the task it is executing. However, if one of the fields you want checked contains an invalid entry, the action server or Tx link puts the story in an error queue and ignores the remaining commands in the task.

## Background and Possible Uses of Validation

Validation is designed to help users attach accurate distribution codes to stories before sending them to a distribution server. See “Distribution Servers” on page 14-30 for more information. Part of its function is to combine entries in each field it checks into a distribution code. If all fields contain acceptable values, it attaches that code to the story. You can have the action server or Tx link use a `move` or `dup` command to overwrite this code with another one.

See “Assigning Distribution Codes” on page 14-32 for more information.

This way, a user can put a code the user wants to attach to a story in one of the story’s fields. The user can send the story to an action server or Tx link that is set up to check that field. If the field contains a valid distribution code, the action server or Tx link attaches that code to the story and sends the story to a distribution server. Otherwise, it sends the story to an error queue and notifies the user of the error.

You can also use validation to allow people to enter different parts of a distribution code in different fields. If the action server or Tx link determines that these fields contain acceptable values, it combines the contents of the fields into a distribution code, which it attaches to the story before sending it to a distribution server.

Though validation was designed primarily to help create accurate distribution codes, you can use it anywhere you need to have an action server or Tx link verify the contents of one or more of a story's fields before processing the story. For instance, you could use validation to have an action server check the created by and modified by fields in each story it processes to ensure that it accepts only stories created or modified by certain users.

It could also be used to rid the rundown of unwanted pages before archiving by searching for a specific string in a field and removing any stories with that string. Any stories the producer did not want or need to archive could be coded with that string and later stripped from the rundown by a timed-interval action server using validation.

## Using Validation with Action Servers or Tx Links

You can use validation with an action server or a Tx link. Adding validation to either uses the same process.

In the example, we set up the Tx link to send stories marked with a distribution code that indicates whether they are news or sports stories and whether they are intended for television or radio. First, create a form that writers can use to mark stories as either news or sports stories and indicate whether they are intended for television or radio.

The writer indicates whether the story is for television or radio by typing either `tv` or `radio` in the `TV/RADIO` field. Likewise, the writer indicates whether the story is a news or sports story by typing either `news` or `sports` in the `NEWS/SPORTS` field.

The Tx link that we set up checks each of these fields. If it determines that both contain valid entries, it combines the contents of these fields

into a distribution code. Then it attaches that code to the story and sends the story to the affiliate.

## Using the Validation Feature

To use the validation feature with a server, create a validation story that contains the instructions you want the server to use for validation. This story can be in any queue (such as `SYSTEM.VALIDATION`).

The validation story consists of one or more verification instructions. Each verification instruction identifies a field you want the server to check and specifies a list of values that field is allowed to contain. The format for a verification instruction is:

```
verify <field name> <length> [ignore]
<list of allowed values for the field>
```

1. Begin each verification instruction with the word `verify` followed by the character that identifies the type of field you want to check. For instance, put `title` here to check a slug field.
2. Specify the number of characters you want to check in that field. For instance, to check three characters, type `3`. The server compares only the first three characters that appear in that field against the list of acceptable values for the field.

This number also controls how many characters in a field are used in the distribution code. If you specify three, only the first three characters in the field are used. The server converts any space characters (trailing or leading) to hyphens so that it can add them to the distribution code.

3. (Optional) Use the `ignore` job list command in an instruction to indicate, if the field is validated, the server should skip validation of the other fields and build the distribution code as if all fields contained acceptable values.

If the field fails verification, the server moves to the next field to be verified and does not consider the story to be invalid. You can use `ignore` only once in a validation story.

#### 4. List the values you allow for the field.

If you list more than one value, separate them with spaces. List as many values on as many lines as necessary. Include + and - characters to indicate ways in which you want the field to be checked.

- A plus sign (+) means the field can be blank.
- A minus sign (-) means the field can be any value *except* blank.
- A plus and minus sign (+ -) used together means that any value is acceptable for the field. This is not the same as using the ignore option on that field.

#### 5. Set the length for each field to the maximum value expected in that field (the length of radio and sports, respectively).



*For clarity, provide more characters for each item in the list of acceptable values for a certain form than you have specified in the verify line. The server ignores the extra characters. You can list possible values along with the + - option, which tells the server to accept anything in that field as valid. Any values you list in addition are superfluous, but harmless.*

## Validation Job List Commands

An action server or Tx link has a job list that contains a separate task for each queue you want it to watch. If you want an action server or a Tx link to validate fields, you must put the appropriate validation commands in each task of the job list where you want it to check the fields.

The validation commands must appear after the task's scan line and before any other commands. You can have up to three validation commands in a task. The first two are optional. The last one is required if you want the Tx link or action server to validate fields in that task.

```
quiet no
```

```
ignore yes
```

```
validate <validation queue name> <error queue name>
```

These job list commands are described as follows:

`quiet no`

An action server or Tx link that validates fields displays a message only if it encounters a field that does not contain an acceptable value. Otherwise, it processes the story. If you include the `quiet no` validation instruction in the task, the action server or Tx link also displays a message indicating success whenever it successfully validates a story.

`ignore yes`

Occasionally, you may want to create a task that builds distribution codes from the contents of certain fields, but does not validate those fields. Include the `ignore yes` validation instruction in the task. Then, the action server or Tx link accepts whatever appears in the fields you have it check.

`validate <validation queue name> <error queue name>`

Include this instruction in the task for the action server or Tx link to validate fields when it executes the task. This instruction must begin with the word `validate` followed by the queue containing the validation story that you want the server to use with that task. Then, list the error queue where you want the action server or Tx link to place stories that it cannot validate.

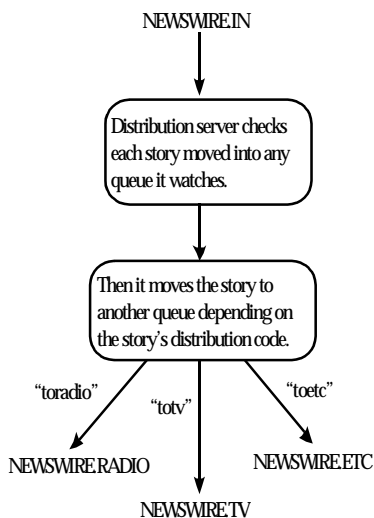
If you use either `ignore yes` or `quiet no`, they must precede this instruction.

## Distribution Servers

A distribution server is used to route stories based on their *distribution codes*. Distribution codes are text strings of up to 32 characters which you can assign to your stories. A distribution code that is longer than 32 characters is truncated to 32 characters before being assigned to the story. When you assign a distribution code to a story, you overwrite any distribution code that story may already have.

Distribution servers are usually used by large stations or networks who write their own wire stories and send them to their affiliates. By attaching distribution codes to these stories, their affiliates can distribute the stories to appropriate queues.

Distribution servers can also monitor a queue. When a new story is added to a queue, the distribution server will move the story to another queue based on the distribution code and instructions you assign.



**Figure 14-3 Distribution Server**

Also, distribution instructions in a distribution story can use the same `always`, `silent`, `urgent`, `flash`, and `bulletin` options that wire service distribution instructions use. These options must appear at the end of the distribution instruction if you use them.

## Assigning Distribution Codes

To create a distribution instruction that matches more than one distribution code, use a wildcard. A wildcard is a special character (a pound sign (#) in this case) that matches any character.

For instance, the following distribution instruction tells the distribution server to match any code that begins with “tv” and send the story to AVSTARWIRE.TVNEWS:

```
tv#      avstarwire.tvnews
```

When the wildcard character is placed at the end of the pattern, the distribution server interprets it as an instruction to match all remaining characters. If it occurs anywhere else in the pattern, the distribution server matches only characters that occur in that position.

tv#	Matches anything that begins with tv.
AP#a	Matches any code that begins with AP and contains a in the fourth position.

When the distribution server is trying to match a story’s code to an instruction line, it stops at the first line that matches. It then looks for only those lines in the story that have a pattern exactly like that line.

Suppose you have the following two instructions, and the distribution server is trying to match a story with distribution code tvsports.

```
tv#      avstarwire.all
tvsports avstarwire.tvsports
```

The distribution server matches the first instruction that contains the pattern tv#, and then it looks for any other line that also contains the pattern tv#. In this case, there is no such line, so it sends the story to AVSTARWIRE.ALL and not to AVSTARWIRE.TVSPORTS.

If you use just a wildcard character as a distribution code, that instruction should come at the end of the job list story. For instance, to create



an instruction that handles any stories with distribution codes that do not match any other distribution instructions. Once the server matches a wildcard only, it stops looking for additional matches for the story.

## Using Wildcards and the Destination Queue

You can also use a wildcard as a part of a distribution instruction's destination queue. Whenever the distribution server uses that instruction line, it replaces the wildcard with the distribution code attached to the story it is processing.

For instance, suppose that your distribution story has the following instruction line:

```
tvnews    avstarwire.#
```

In this case, whenever it processes a story with the distribution code `tvnews`, it uses this line. In doing so, it uses the story's distribution code in place of the wildcard in the destination queue and distributes stories with the distribution code `tvnews` to `AVSTARWIRE.TVNEWS`.

This is most useful when you also use a wildcard in the instruction's distribution code. You can create an instruction that handles stories with several different distribution codes.

For instance, suppose you want your distribution server to send stories with the distribution code `tvnews` to `AVSTARWIRE.TVNEWS` and stories with the distribution code `tvsports` to `AVSTARWIRE.TVSPORTS`. You can do this with a single instruction if you use a wildcard in the instruction's distribution code and destination queue:

```
tv# avstarwire.#
```

This instruction causes the server to copy any story with the distribution code `tvnews` to `AVSTARWIRE.TVNEWS`. Likewise, it causes the distribution server to copy any story with the distribution code `tvsports` to `AVSTARWIRE.TVSPORTS`.



*If you try this, ensure that what gets substituted into the wildcard portion of the destination queue forms the pathname of an existing queue. Otherwise, the server cannot copy stories to that queue and sends them to `SYSTEM.UNKNOWN`, or the default error queue you named.*

## From the Command Line

You can use the `move` and `dup` commands to attach a distribution code to a story when you send it to a distribution server. If people on your system use these commands frequently to attach distribution codes to stories, you can publish the codes that you want them to use.

## Using an Action Server or Tx Link

Tx links are explained in “Networking Two or More Servers Using Rx/Tx Links” on page 14-88.

You can attach distribution codes to stories that you send to a distribution sever using an action server or a Tx link. There are two ways to use action servers and Tx links to do this.

- Dup or Move commands
- Validate commands

### Using Dup or Move Commands in Job Lists

You can use the `move` and `dup` commands in action server and Tx link job lists to attach distribution codes to the stories those servers process. For instance, the following job list shows how you may have an action server attach a distribution code to any story added to `TVNEWS.OUT`:

```
scan tvnews.out
    dup avstarwire.send tvnews
```

The job list tells the server to attach the code `tvnews` to each story it handles and then copy the story to `AVSTARWIRE.SEND`.

Use this method to automatically assign a particular distribution code to every story added to a particular queue. If you need more flexibility in how the server assigns codes, see “Assigning Distribution Codes” on page 14-32.

## Using Validate Commands in Job Lists

Action servers and Tx links support a feature called *validation*. This feature allows you to have an action server or Tx link extract a distribution code from a story's form and check whether it is a valid code. If the code is valid, the server attaches it to the story and sends the story to the distribution server. If the code is invalid, the server puts the story in an error queue and notifies the user. Validation is covered in more detail in "Assigning Field Validation" on page 14-26.

## Instructions in the Wire Distribution Story

A wire program relies on instructions in the wire distribution story to determine what to do with each story it receives. This allows the wire program to use each story's wire service code to determine where to send the story.

Each wire program can have no more than 150 distribution instructions. If you need to handle more than 150 codes from a particular wire service, you can use a distribution server to get around this limit.

Add special distribution instructions to the wire distribution story. These instructions cause the wire program to look for stories with the wire service codes that you want the distribution server to handle. The wire program assigns each story a distribution code that matches their wire service code and sends them to the distribution server's source queue.

Those instructions should match the following format. They should begin with a wire service code that you want the distribution server to handle. Follow the code with the distribution server's source queue. You must follow the destination queue with the `TRANSMIT` option:

```
<wire service code> <destination queue name> TRANSMIT
```

When a story comes in with that wire service code, the wire program gives the story a distribution code that matches its wire service code. Then the wire program sends the story to the destination queue specified in the instruction.

You can use options such as `ALWAYS`, `SILENT`, and `URGENT` with the `TRANSMIT` option. A distribution instruction that assigns stories distribution codes matching their wire service codes does not have to send those stories directly to a distribution server. The instruction can send them to an intermediate queue and then an action server can send them to the distribution server.

## Matching and Case

When using two or more distribution instructions that begin with the same code, you must enter the codes in the same case. If you do not, the distribution server does not execute all the instructions when it receives a story with that distribution code.

When a distribution server processes a story, it begins by checking whether the first instruction in its distribution story matches the story's distribution code. If not, the server checks the second instruction, and then the third, and so on, until it finds one that matches the story's distribution code.

When it finds a match, it sends the story to the queue specified in that instruction. Then the distribution server does a case-sensitive search on the remaining instructions in the distribution story to see whether any use exactly the same distribution code (that is, the same code entered in the same case) as the instruction that it has matched. If any instructions match, the server sends the story to the queues specified in those instructions as well.

For instance, if a distribution server successfully matches an instruction with a distribution code of `tvnews`, it looks for other instructions that also begin with `tvnews` and sends the story to the queues specified in those instructions. It ignores instructions that have that code in a different case, such as `TvNews` or `TVnews`.

## Matching and Order

Usually, the order of instructions in the distribution story does not matter. It becomes important when you have a distribution code that

also forms the beginning of another distribution code, such as `tv` and `tvnews`. In this case, you need to make sure that instructions that use the longer code appear before those that use the shorter code.

When the distribution server attempts to match an instruction's code to a story's distribution code, it considers that it has a match if the instruction's code matches some initial portion of the story's distribution code. For instance, the distribution server would match an instruction for code `tv` to a story with distribution code `tvnews`.

Once the distribution server has matched an instruction code to the story, it stops trying to match additional instructions to the story. Instead, it looks for other instructions that have the same code as the story that it has matched. It never matches the instructions with the longer codes.

If you had a distribution story with the following instructions in the order shown below, and if you were to send a story with distribution code `tvnews` to this distribution server, the distribution server would match the first instruction in the distribution story but would never execute the second:

```
tv          newswire.tv
tvnews     newswire.tvnews
```

In this example, the server always matches the first instruction to any story that has a distribution code that begins with `tv`. Then it looks for other instructions that have `tv` as their code. It never executes the instruction that has the code `tvnews`. Always arrange such instructions so that those with longer codes come before those with shorter codes. That way, only stories with the distribution code `tvnews` (or larger) match the first line. Stories with the code `tv` fall through the first line and match the instruction in the second line.

## Adding a Distribution Server

To set up a distribution server to watch a queue and distribute stories placed in a queue, do the following:

1. Choose a device and mailbox number for the server which will also be used for associated queues. See “Adding a Server Program to the System” on page 14-3 for more information.
2. Add the server (utility program) to the configuration file on each Avstar Server—such as servers A and B—in your system. Details for this step are provided in “Adding a Server Program to the System” on page 14-3.
3. Create a job list queue in the Configure folder of the System directory in the Avstar database. The queue’s name, such as 257, is the number selected in step 1. For instance, the queue’s full pathname is `SYSTEM.CONFIGURE.257`.



*You can end the name with a hyphen and descriptive word, such as “distribution.” The queue’s full pathname would then look like:*

*`SYSTEM.CONFIGURE.257-DISTRIBUTION`. See “Creating a New Queue” on page 5-6 for more information.*

4. Create a job list story for the server, by doing the following.
  - a. Open the job list queue, such as `SYSTEM.CONFIGURE.257`.
  - b. Create a new story, the first in the job list queue, to hold the distribution server’s job list. See “Creating a New Story” on page 5-9 for more information.
5. Add tasks that you want the distribution server to perform; include a separate task for each queue the server watches. Each task in the job list story consists of two lines with the following format:

```
source <queue scanned by server>
distribution <distribution queue> [<error queue>]
```

**source**

Identifies which queue the distribution server should scan. This job list command is the same as the `scan` command, used by action servers. The queue is known as both the scan queue and source queue.

If you do not specify an error queue, the distribution server sends any such stories to `SYSTEM.UNKNOWN`.

**distribution**

Specifies the queue containing the distribution story and, optionally, an error queue for stories whose distribution codes cannot be processed.

For instance, the job list story may appear like this:

```
source newswire.in
distribution system.distribution.257 newswire.error
```



*Unlike action servers where the scan queue identifier (scan line), tasks, and commands (instructions) are located altogether in the job list story, a distribution server separates the instructions (except for the distribution command), placing them in another queue and story, called the distribution queue and distribution story, respectively. The distribution queue is typically given the same numerical name as its corresponding job list queue and is located in the `SYSTEM.DISTRIBUTION` directory.*

6. Create the distribution queue and story, by doing the following:
  - a. Navigate to `SYSTEM.DISTRIBUTION`. (If this directory does not exist, create it. See “Creating a New Directory” on page 5-4 for more information.)
  - b. Create a new queue, giving it the same numerical name as the job list queue for the distribution server, such as 257. The full pathname for the new queue would be `SYSTEM.DISTRIBUTION.257`. See “Creating a New Queue” on page 5-6 for more information.
  - c. Open the new queue and create the distribution story.

Each instruction in the distribution story is one line long and has the following format:

```
<distribution code> <destination queue>
```

For instance, suppose we know that some stories sent to our distribution server have `tvnews` as their distribution code and others have `tvsports`. We want the distribution server to send stories with the `tvnews` code to

```
AVSTARWIRE.TVNEWS and those with tvsports to  
AVSTARWIRE.TVSPORTS.
```

In this case, you would type the following two instructions in the distribution story:

```
tvnews avstarwire.tvnews  
tvsports avstarwire.tvsports
```

If you want to send stories that have a certain distribution code to more than one queue, follow this example:

```
tvnews avstarwire.news  
tvnews avstarwire.all  
tvsports avstarwire.sports  
tvsports avstarwire.all
```

7. Assign mailbox database traits, using the number matching the job list queue's name and server's device number chosen in step 1, to appropriate source queue(s), as identified in step 5. For instance, the job list queue is named 257, so that is the standard mailbox number assigned to queues identified in the job list associated with the distribution server.
  - a. Open the Queue Properties dialog box for a scan queue, such as `NEWSWIRE.IN`. See "Changing Database Traits" on page 5-21 for more information.
  - b. Click on the Maintain tab. See "Maintain Tab" on page 5-38 for more information.
  - c. Select the Standard radio button.
  - d. Type in the mailbox number, such as 257.



- e. Click OK to save changes.
8. Repeat step 7 as needed for each source queue in the job list.
9. (Optional) Test your configuration changes. See “Testing the Site Configuration File After Changing” on page 11-9 for more information.
10. Reconfigure the system.
11. Start the distribution server. For instance, to start a distribution server with device number 257, type:
 

```
restart 257
```
12. (Optional) Back up site files using `sitedump`.

Details for steps 10-12 are provided in “Adding a Server Program to the System” on page 14-3.

## Parallel Wire Servers

Another use for a distribution server is to help provide backup wire distribution in case the usual wire distribution fails—if, for instance, the PCU port receiving the wire goes down. This is accomplished by a distribution server working in tandem with a parallel wire server.

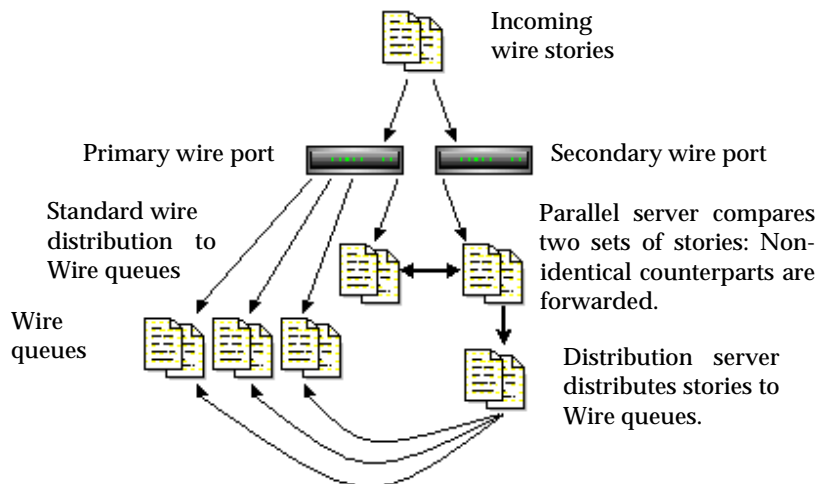


Figure 14-4 Parallel Wire Servers

A parallel wire server works as follows:

- The same wire service is received through two different PCU ports: preferably on two different PCUs on two different Avstar Servers. One of these ports uses the standard wire distribution feature, with a line added to distribute its wires to the parallel wire server. The second port uses one line in the distribution story, which routes its wire stories to the parallel wire server.
- The parallel wire server checks to make sure that each story received from the secondary wire has a counterpart from the primary port. If a secondary wire story cannot be matched, it is passed on to a distribution server. No action is taken on unmatched primary wire stories, since they were already distributed normally.
- If the primary wire port fails, each secondary story your system receives lacks a corresponding primary story, so the parallel wire server passes each one to a distribution server. This server uses a distribution story nearly identical to the wire distribution story, so stories are routed to the same queues as with standard wire distribution. Wire reception thus appears normal to users.

Although using two PCU ports for the same wire may seem like a waste of a port, this redundancy can be a lifesaver in the long run. However, a parallel wire server is useful only for problems within the system such as a PCU port or computer problem. If the reception problem is in phone lines, satellite antenna, or the wire service distribution box, receiving wire stories on two PCU ports does not help.

### **Database Components for Backup Wire Distribution**

Only one set of parallel and distribution servers are necessary, even if you are providing backup for several wires. For backup wire distribution to work, distribution and parallel servers require certain database components, which are:

- A pair of scan queues to hold incoming wire stories for comparison by the parallel server—typically `SYSTEM.PARALLEL.A1` and `SYSTEM.PARALLEL.A2`.

- A queue where the parallel server forwards unmatched stories for the distribution server to scan—typically `SYSTEM.PARALLEL.WIRES`
- A queue containing the wire distribution story for the secondary port, should it need to take over wire distribution—typically `SYSTEM.PARALLEL.DISTRIBUTION`
- Queues containing job lists used by the parallel and distribution servers located in the `SYSTEM.CONFIGURE` directory.

## Adding a Parallel Wire Server

A parallel wire server provides duplicate copies of a wire in several PCUs. If one PCU or wire is offline, the system will turn on the server in the other PCU or wire to continue supplying stories from the wire. For your parallel wire server to be truly redundant, each wire of the parallel server must be connected to an independent source of the incoming service.

Here, we assume that your site requires continuous reception of the AP wire, so we set up a parallel wire server to receive stories from that wire service.

To add a parallel wire server to the Avstar system, do the following:

1. Choose a device and mailbox number for the parallel server which will also be used for associated queues. See “Adding a Server Program to the System” on page 14-3 for more information.
2. Add the server program (`server` line) to the configuration file on each Avstar Server—such as servers A and B—in your system. Details for editing the configuration file in this step are provided in step 2 of “Adding a Server Program to the System” on page 14-3.

However, additional editing of the configuration file (`/site/config`) is required for parallel servers.

Wires using the `anpa7` program to interpret incoming stories, such as AP, restrict device names to two characters. A1 and A2 are used in this procedure.

See “PCU Device Numbering” on page 11-34 for more information.

- a. You must determine which PCU port is free for use as the second wire port. If your main AP wire is on port 4 of PCU 10, it would probably look like this in the configuration file:

```
wire 14 1200 anpa7 AP - ;AP wire
```

Change the device name AP to A1 to avoid name duplication, and change the comment field at the end of the line to help identify it as the primary wire port. It should look similar to this:

```
wire 14 1200 anpa7 A1 - ;primary AP wire
```

- b. Choose an unused port on a different PCU running on a different Avstar Server for the secondary wire feed. For instance, if PCU 10 (with its AP wire on port 4) is running on server A, you should select a PCU running on server B, such as PCU 20. If port 6 is available on PCU 20, the server program’s device number, 26, is added to PCU 20’s configuration line:

```
ccu 20 pcu20 pc 21 22 23 24 25 26
```

- c. Then, the secondary wire line is added, which should look like this:

```
wire 26 1200 anpa7 A2 - ;secondary AP wire
```

- d. Save the configuration file according to step 2 of “Adding a Server Program to the System” on page 14-3.



*Do not reconfigure or restart the wire programs yet. They do not match the codes in `SYSTEM.WIRES.DISTRIBUTION`, so all wires would be sent to `WIRES.UNKNOWN`.*

3. Log in to an Avstar Workstation. (You must create several queues and stories, as well as modify some that may already exist.)
4. Create a folder (directory) called Parallel in the System directory. The full pathname for this folder is `SYSTEM.PARALLEL`. See “Creating a New Directory” on page 5-4 for more information.

5. In `SYSTEM.PARALLEL`, create four new queues, giving them the following names: A1, A2, Wires, Distribution. The full pathnames for these queues will be:

`SYSTEM.PARALLEL.A1`

`SYSTEM.PARALLEL.A2`

`SYSTEM.PARALLEL.WIRES`

`SYSTEM.PARALLEL.DISTRIBUTION`

See “Database Components for Backup Wire Distribution” on page 14-42 for more information on the purpose for each of these queues. Also, see “Creating a New Queue” on page 5-6 for information on how to create a queue.



*When wires are being distributed normally, the parallel server immediately removes matching stories from A1 and A2 queues, so purging the queues is not necessary. If the primary or secondary wire input stops reaching the A1 or A2 queue, automatic removal of stories from either queue stops. If the parallel server (or the Avstar Server it is on) dies, both queues continue to fill up until you run out of space. Finally, if the distribution server dies, `SYSTEM.PARALLEL.WIRES` fills up.*

*To prevent these problems, set a short purge interval (less than six hours) for these three queues: A1, A2, and Wires. See “Choosing Queues to be Purged” on page 5-40 for more information.*

6. Assign mailbox database traits to three of the new queues, according to the following criteria:
- `SYSTEM.PARALLEL.A1` – shares same mailbox number as parallel server, such as 258
  - `SYSTEM.PARALLEL.A2` – shares same mailbox number as parallel server, such as 258
  - `SYSTEM.PARALLEL.WIRES` – shares same mailbox number as distribution server, such as 257

Mailbox numbers must match the server’s device number, as indicated in the configuration file. Otherwise, changes in the queue

will not activate the correct server program. To assign the mailbox database trait to a queue, do the following:

- a. Open the Queue Properties dialog box for a queue, such as `SYSTEM.PARALLEL.A1`. See “Changing Database Traits” on page 5-21 for more information.
  - b. Click on the Maintain tab. See “Maintain Tab” on page 5-38 for more information.
  - c. Select the Standard radio button.
  - d. Type in the mailbox number, such as 258.
  - e. Click OK to save changes.
  - f. Repeat steps a-e until queues have assigned mailboxes.
7. Create the parallel server’s job list queue in the `SYSTEM.CONFIGURE` directory in the Avstar database. The queue’s name, such as 258, is the number selected for the parallel server in step 1, and should match the number chosen for mailboxes in step 6. The queue’s full pathname is `SYSTEM.CONFIGURE.258`.



*You can end the name with a hyphen and descriptive word, such as “parallel.” The queue’s full pathname would then look like:  
`SYSTEM.CONFIGURE.258-PARALLEL`.*

8. Create a job list story with tasks for the parallel server.
- a. Open the job list queue, such as `SYSTEM.CONFIGURE.258`.
  - b. Create a new story—the first in the job list queue—to hold the parallel server’s job list. See “Creating a New Story” on page 5-9 for more information.

- c. Add tasks that you want the parallel server to perform. Each task in the job list story consists of three lines, in the following format:

```
local <queue scanned by server>
remote <queue scanned by server>
distribution <destination queue>
```

The commands, `local` and `remote`, do not describe actual queue locations; both queues are on the same system.

`local`

Identifies which queue contains wire stories forwarded to the parallel server through normal wire distribution. This job list command is similar to the `scan` command, used by action servers. The parallel server will scan the local queue and compare its contents to those in the remote queue.

`remote`

Identifies which queue contains the wire stories received through the secondary wire port. This job list command is similar to the `scan` command, used by action servers. The parallel server will scan the remote queue and compare its contents to those in the local queue.

`distribution`

Specifies the destination queue where the parallel server forwards non-identical stories—the source queue scanned by the distribution server.

For instance, the job list story may appear like this:

title	presenter
Parallel wire server	
local	system.parallel.a1
remote	system.parallel.a2
distribution	system.parallel.wires

If the distribution server has not been set up, you must create it as well as its job list queue and story. See “Adding a Distribution Server” on page 14-38 for more information.

Your system’s wire distribution queue and story were included when your system was installed.

9. Modify the distribution server’s job list story.

For instance, go to `SYSTEM.CONFIGURE.257-DISTRIBUTION` and add the following lines to its job list story:

```
source      system.parallel.wires
distribution system.parallel.distribution
```

10. Since you want the parallel server to distribute stories in the same way as standard wire distribution, copy the system’s wire distribution story (in `SYSTEM.WIRES.DISTRIBUTION`) to the `SYSTEM.PARALLEL.DISTRIBUTION` queue.

- a. Open the first story in `SYSTEM.WIRES.DISTRIBUTION`.
- b. Click the Edit drop-down menu.
- c. Select Copy To.
- d. Enter `SYSTEM.PARALLEL.DISTRIBUTION`, the destination queue.

11. Back up the original wire distribution story in the `SYSTEM.WIRES.DISTRIBUTION` queue by making a copy of it in a different queue.

12. Return to `SYSTEM.WIRES.DISTRIBUTION` to make the following modifications:

- a. Modify the *bottom* story (Leave the *top* copy unchanged, for now) so your system’s wire distribution sends a copy of every incoming wire story to the two queues scanned by the parallel server (`SYSTEM.PARALLEL.A1` and `SYSTEM.PARALLEL.A2`) as specified in the parallel server’s job list story. To do this, add two lines (as shown) below the line that sends a copy of each wire story to `WIRES.ALL`. It should look like this:

AP#####	wires.all	ALWAYS	URGENT
A1#####	system.parallel.a1	ALWAYS	
A2#####	system.parallel.a2	TRANSMIT	



- b. Modify every line in the wire distribution story that starts with AP, changing AP to A1. This makes it possible to tell whether a given wire story came from standard wire or parallel server distribution. When you are done, the three lines in the story should look like this:

A1#####	wires.all	ALWAYS	URGENT
A1#####	system.parallel.a1	ALWAYS	
A2#####	system.parallel.a2	TRANSMIT	



*Do not change the last line's A2 to A1. The A2 indicates the line that transmits stories to the SYSTEM.PARALLEL.A2 queue.*

- c. Save the story.
- d. Move the modified distribution story (at the bottom) to the top of the queue, using the `order workstation` command; this will make it the primary wire distribution story.
13. Connect the secondary wire to the PCU port.

Before you can receive wires on the secondary wire port, ensure there is a physical connection to provide those wires. Details vary depending on the hardware, but in the example you must ensure a wire service line is connected to the PCU port you listed in the configuration file (26 in the example). See step 2a on page 14-43.

This wire service line could come from a DR400 box, an Equatorial box, or some other kind of signal splitter or multiplexor. Ensure cabling and connectors are correct. See your hardware installation manual for details.

14. Reconfigure the system. See “Adding a Server Program to the System” on page 14-3 for more information.
15. Restart the original wire port, and the parallel and distribution servers.

You also need to restart the PCU using the new wire program, such as PCU20.

Ensure you select the Avstar Server the original wire port, servers, and PCU are configured on.

AVSTAR-A: **restart 257**

AVSTAR-A: **restart 258**

AVSTAR-B: **restart 20**

You will receive `Hot-to-go` messages for the devices on PCU 20. Also, a message similar to the following will appear:

```
S258: Sat Jun 3 08:22:13 2000 matching system.parallel.a2\
```

The message is from the parallel server and appears when it is started, indicating the server is alive and comparing the standard wire reception with wire stories routed to the A2 queue.

If primary reception fails, the parallel server gives the following message when it wakes up the distribution server:

```
S258: Sat Jun 3 08:22:13 2000 distributing system.parallel.a2
```

This message is the only way to tell when the parallel server is forwarding secondary wire stories for distribution. You can tell if these stories are being distributed by A2 instead of A1 by checking the wire-distribution codes in the stories.



*There may not be any indication if secondary wire distribution fails. Check the local queue (`SYSTEM.PARALLEL.A1`) occasionally to see if it is accumulating stories. If so, this would indicate a problem with reception of the secondary wire.*

If there is a problem with something other than the primary wire PCU port—for instance, if the wire fails in the phone or satellite connection—the redundancy provided by the parallel wire server cannot provide continuous wire service reception.

## Keyword Servers

A *keyword server* is a utility program that searches stories it processes for specific keywords. It distributes copies of each story to different queues, based on which keywords the story contains. Unlike wire and telex programs, which are limited to searches that contain no more than 250 unique keywords, a keyword server can perform searches that contain as many as 1000 unique keywords.

When you install a keyword server, it runs on one of your system's Avstar Servers, not on a PCU. While this allows the keyword server to handle long lists of search rules, it also means the keyword server may affect your system's performance.

The impact a keyword server has on your system depends on what kind of equipment your system uses and how much of a load it is carrying. In some cases, a keyword server will not have any impact at all. In others, there may be a noticeable reduction in performance. If you find that a keyword server degrades your system's performance too much, reduce the amount of searching the keyword server does.



**Since keyword servers could affect your system's performance, use keyword servers only when wire capabilities are inadequate.**

Suppose we have set up a wire program to perform keyword searching on stories it receives from the Associated Press (AP). Wire programs can only use 250 unique keywords in searches they perform.

To get around this limitation, a keyword server may be created to help the wire program, which still searches each story it receives using its own search rules. However, it also sends a copy of each story it receives to the keyword server, which in turn searches each copy using its own search rules. This effectively increases the number of unique keywords allowed to 1250: 1000 for the keyword server and 250 for the wire program.

An instruction is required in the wire distribution story that has the wire program duplicate a copy of each story it receives to a queue, which the keyword server watches. Some modification of the wire program's keyword job list queue (`SYSTEM.WIRES.KEYWORDS`) will also be required.

### **Database Components for Expanded Wire Keyword Searches**

For expanded wire keyword searches to work, the keyword server requires certain database components. These components are:

- Queue containing job list used by the keyword server. It must be located in the `SYSTEM.CONFIGURE` directory.
- A source queue to hold incoming wire stories for scanning by the keyword server—such as `WIRES.AP-SEARCH`.
- A queue—typically named `SYSTEM.WIRES.KEYWORDS`—containing the search job list story. This queue is used by the system's standard wire searching feature; therefore, it should already exist in your database.
- Queues containing stories with search rule sets of keywords and destination queue names. The maximum number of search rule queues are 20 per keyword server.
- Destination queues where the keyword server forwards stories/search results.
- A queue containing the wire distribution story—typically named `SYSTEM.WIRES.DISTRIBUTION`. This queue is used by the system's standard wire distributing feature; therefore, it should already exist in your database.

Here is a sample workflow diagram for a keyword server.

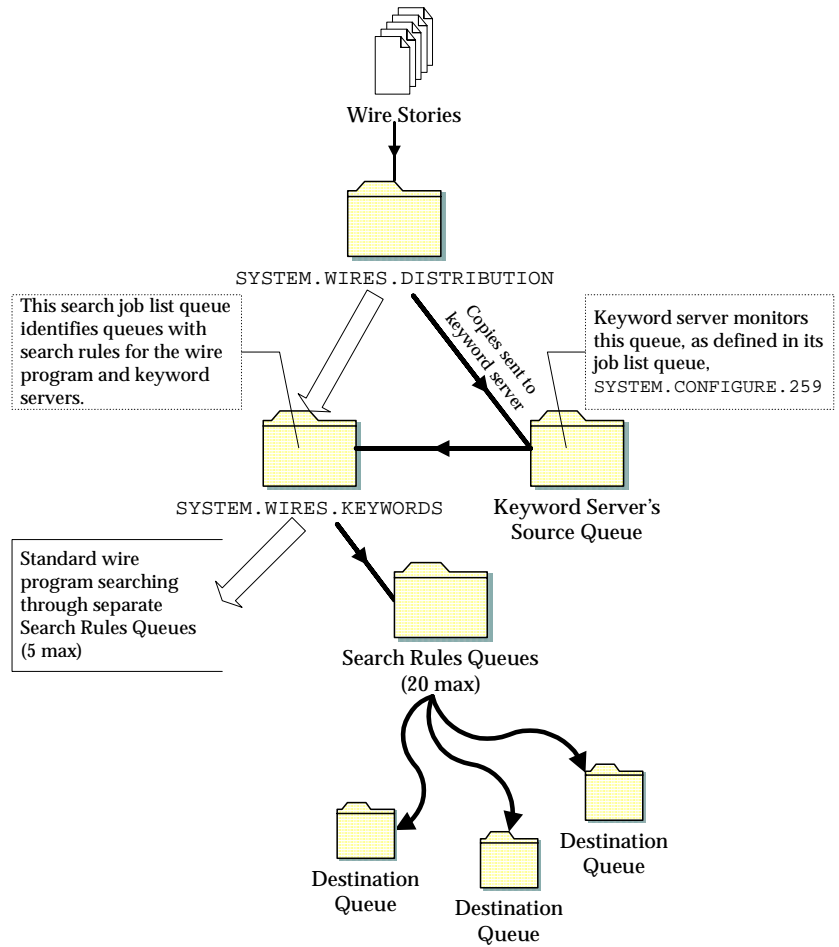


Figure 14-5 Keyword Server Workflow

## Adding a Keyword Server

To add a keyword server to the Avstar system, do the following:

1. Choose a device and mailbox number for the keyword server which will also be used for associated queues. A device name should also be chosen, such as `key1`; it will be needed for the next step in the procedure. See “Adding a Server Program to the System” on page 14-3 for more information.
2. Add the server (utility program) to the configuration file on each Avstar Server—such as servers A and B—in your system. Details for editing the configuration file in this step are provided in step 2 of “Adding a Server Program to the System” on page 14-3.



*In addition to adding the keyword server to the standard host definition of the Avstar Server on which you want it to run, ensure you also add it to each alternate host definition.*

*A keyword server’s device name, such as `key1`, is used in the search job list; ensure that you always include a device name when adding a keyword server to the system. For instance, the configuration line may appear like this:*

```
server 259 keyword 259 key1 ;keyword server
```

*See step 2 of “Adding a Server Program to the System” on page 14-3 for more information.*

3. Create a job list queue in the Configure folder of the System directory in the Avstar database. The queue’s name, such as 259, is the same number selected in step 1. For instance, the queue’s full pathname is `SYSTEM.CONFIGURE.259`.



*You should end the name with a hyphen and descriptive word, such as “key1.” The queue’s full pathname would then look like: `SYSTEM.CONFIGURE.259-KEY1`. See “Creating a New Queue” on page 5-6 for more information. In this example, the keyword server’s device name is used as the descriptive word for the job list queue name. The device name is an important factor of later steps in this procedure.*

4. Create the keyword server's job list story in the database, by doing the following:
  - a. Open the job list queue, such as  
SYSTEM.CONFIGURE.259-KEY1.
  - b. Create a new story, the first in the job list queue, to hold the keyword server's job list. See "Creating a New Story" on page 5-9 for more information.
  - c. Add tasks that you want the keyword server to perform; include a separate task for each queue the server watches. Each task in the job list story consists of only one line, with the following format:

```
source <queue scanned by server>
```

```
source
```

Identifies which queue the keyword server should scan. This queue is the same queue to which the wire program will be instructed to send wire story copies.

For instance, here is an example of a job list story for a keyword server that scans WIRES.AP-SEARCH when it gets a wake-up notification:

title	presenter
joblist for keyserver	
;this server searches wires duped to wires.ap-search	
source	wires.ap-search

A single task is defined in the keyword server's job list, with a commented line—a line preceded with a semicolon (;)—offering further details about the keyword server.



*When the keyword server processes a story, it deletes it from the source queue. You do not need to clear old stories out of the keyword server's source queue. It is recommended that you "hide" the source queue from general user accounts, by assigning it a read group, such as sysops. See "Groups Tab" on page 5-34 and "Read Group" on page 6-23 for more information.*

5. Assign the mailbox database trait, using the number matching the job list queue's name and server's device number chosen in step 1, to the appropriate source queue(s), as identified in step 4c. For instance, the job list queue is named 259, so that is the standard mailbox number assigned to queues identified in the keyword server's job list.
  - a. Open the Queue Properties dialog box for a scan queue, such as `WIRES.AP-SEARCH`. See "Changing Database Traits" on page 5-21 for more information.
  - b. Click on the Maintain tab. See "Maintain Tab" on page 5-38 for more information.
  - c. Select the Standard radio button.
  - d. Type in the mailbox number, such as 259.
  - e. Click OK to save changes.
6. Repeat step 5 as needed for each source queue in the job list.
7. Create the queues and stories to hold search rule sets for the keyword server to use.



*Each keyword server can have 1-20 search rules queues, each with its own keyword story, containing keywords. Each keyword story must be the first one in the queue, and contains a number of keywords equal to 1000 divided by the number of search rule queues. For instance, if 4 queues are set up for a keyword server, each queue's keyword story can hold up to 250 keywords. This evenly distributes the 1000 keywords allowed for keyword servers across search rules queues and their stories. If 10 queues are set up, then each queue's keyword story can have 100 keywords.*

- a. In this sample procedure, two search rules queues are created in the `SYSTEM.WIRES` directory. For instance, the queues' full pathnames could be `SYSTEM.WIRES.KWDSEVER-1` and `SYSTEM.WIRES.KWDSEVER-2`.
- b. In each queue, the first story is considered the *keyword story*, and contains the rule set of keywords.



See “Setting up Keyword Searching” on page 13-39, “Keyword Limitations” on page 13-43, and “Tips on Building Search Rules” on page 13-46 for more information.

The format each rule set must have is as follows:

```
destination <queue name>
<search rule>
[<search rule>]
[<search rule>]
...
```

`destination`

Identifies the queue into which the keyword server will place the results of its keyword search. The name and location of this queue is customizable.

`<search rule>`

Each search rule consists of one or more keywords you want a keyword server (utility program) to use to locate stories on a particular topic.



**Never create a destination line that sends stories to a personal queue or any queue not purged. If a program sends stories to such queues, it can lead to an out-of-space condition.**

Here is an example of what a keyword story and its rule set may look like:

```
destination      wires.keywords.drugs
panama AND drugs
drugs NOT pharmacy OR <drugs AND smuggling>
```

Two rule sets cannot use the same destination queue. Each search rule occupies a separate line in the rule set. Also, the number of keywords allowed (1000 for keyword servers) is defined by the words, not the number of lines. For instance, in the previous

example, four keywords are used—since you do not count each occurrence of a word (just the first), `drugs` is only counted once.



*Each keyword in the story can have a maximum of 10 characters. More examples and detailed information on rule sets and keyword searches are available in Chapter 13. See “Operating Wire Keyword Searches” on page 13-39 for more information.*

8. Add an entry to the first story in the search job list queue, typically named `SYSTEM.WIRES.KEYWORDS`.

a. Go to that queue and open the first story for editing.

b. Insert an entry for the keyword server into the job list, which must begin with `*20`, followed by the keyword server’s device name. If you do not want the program to do any searching, that is all you need to put in the entry. If you want the program to search for keywords, follow the device name with one or more lines that list queues containing search rules you want the keyword server to use.

In step 8b, use the same device name, which was defined in the configuration file in step 2, and the queues with the search rules, which were created in step 7.

For instance, the keyword server has `key1` as its device name and should use search rules in queues:

`SYSTEM.WIRES.KWDSEVER1` and  
`SYSTEM.WIRES.KWDSEVER2`.

Here is an example of how the story may appear.

title	presenter
Keyword indirect	

```

* A1 UPI
system.wires.keywords1
system.wires.keywords2
system.wires.keywords3
;
*20 key1
system.wires.kwdserver1
system.wires.kwdserver2
;
*
system.wires.default1
system.wires.default2

```

9. Modify the wire distribution story—the first story in `SYSTEM.WIRES.DISTRIBUTION`.

For the keyword server to search stories sent by the AP wire service, the wire program must duplicate a copy of each story it receives to the keyword server's source queue. For this to happen, the wire distribution story must be modified.

- a. Open the story and add the new instruction on a *single* line, using the following format:

```
<dev name><wire code> <distribution queue>
[<options>]
```



*There must be no spaces between the device name and the wire service code. The [`<options>`] should also be on the same line, separated from the distribution queue by a space.*

```
<dev name><wire code>
```

For instance, the wire program's device name is `AP`. If you want this instruction to match all wire stories regardless of their wire service code, use `#####` as the code, because it matches any code. See "Creating a Distribution Name" on page 13-32 for more information.

```
<distribution queue>
```

The distribution queue is where you want stories with the specified code to be duplicated, such as `WIRES.AP-SEARCH`.

```
[<options>]
```

Finally, you can end the line with any of the following options: `ALWAYS`, `SILENT`, `URGENT`, `FLASH`, `BULLETIN`, or `TRANSMIT`. For instance, end the instruction with the `ALWAYS` option to ensure that every wire the AP wire program processes is duplicated to the keyword server's source queue.

The finished instruction looks like:

```
AP#####          wires.ap-search      ALWAYS
```

See Table 13-5 on page 13-34 for more information on these options.

Steps 10-13 are described in more detail in “Adding a Server Program to the System” on page 14-3.

10. (Optional) Test your configuration changes.
11. Reconfigure the system.
12. Start the keyword server.

When you see the `System being configured` message, select the Avstar Server where the keyword server is configured and start the keyword server. For instance, type:

```
AVSTAR-A: restart 259
```

When the system displays a `Hot-to-go` message, the keyword server has successfully started.

13. (Optional) Back up site files with the `sitedump` console command.

## System Servers

Utility programs called *system servers* are usually included on your system at installation. You cannot customize them; therefore, they are described here in a limited way. This information should be enough for you to maintain and use these servers, or add them if necessary.

System servers include:

- Seek servers
- Fast Text Search (FTS) server
- Print servers
- Mail servers

## Seek Servers

Seek servers allow users to search for text in the background, while users continue with other program activities, such as script writing. A seek server must be installed on your system.

The seek server program runs on the main Avstar Server, while you continue to work on your client computer (Avstar Workstation). The seek server works by scanning the regular Avstar database, story by story, going through the area you specified in the Find All dialog box—or your seek request story on the Video Terminal (VT). The seek server then returns stories resulting from its search as stories are encountered in the Avstar database.

Seek search is used by anyone who uses Avstar. The seek server can search any queue or folder in your Avstar database. If your site does not have a Fast Text Search (FTS) server, you will use the seek search exclusively.

## Installing a Seek Server

To set up the seek feature on your system:

1. Choose a device and mailbox number for the server program.
2. Add the server program's information to the configuration file.
  - a. The seek server's device number, such as 260, must appear in the `servers` line in the host definition for the Avstar Server, such as server B, on which it runs. The line may appear as follows:

```
servers 257 258 259 260
```

Editing the configuration file requires use of the UNIX line editor. See "Adding a Server Program to the System" on page 14-3, and "Using the UNIX Line Editor" on page 10-4 for more information.



*In addition to adding the seek server to the standard host definition of the Avstar Server on which you want it to run, add it to each alternate host definition.*

- b. Next, create a configuration line for the seek server at the bottom of the configuration file.

A seek server configuration line should look similar to this:

```
server 260 seek 260 SEEK ;seek server
```

- c. After adding information to the configuration file, save your changes.

3. Create the seek queue in the database.

The seek queue's name is defined in the `Q_SEEK` dictionary in the `/site/dict/queues` directory. Usually this name is `SYSTEM.SEK`. Go to your System directory and see if this queue exists. If it does not, create it. See "Creating a New Queue" on page 5-6 for more information.



*This queue must exist for the `seek` command to work. Ensure the write group restriction is not applied so all users can use `seek`. See "Write Group" on page 6-24 for more information.*

4. Assign a form to the Seek queue, `SYSTEM.SEK`.

When users execute the `seek` command, they see the form assigned to `SYSTEM.SEK` on the screen. Among other things, this form has fields where the user can specify what the seek server looks for, where it looks, and where it puts stories that meet the search criteria.

A standard seek form was included with the remainder of your standard forms. The form is stored in `SYSTEM.FORMS.S.SEK`. See "Seek Form" on page 8-31 for more information.

For the `seek` command to use this form, assign it to the Seek queue using the Queue Properties dialog box and its Form tab. See "Changing Database Traits" on page 5-21, "Directory/Queue Properties Dialog Box" on page 5-25, and "Forms Tab" on page 5-27 for more information.

5. Assign a mailbox to the Seek queue.

For the seek server to process a seek request placed in the Seek queue, your system must notify the seek server that a new seek request has arrived there. So, the Seek queue's mailbox number must match the seek server's device number, as indicated in the configuration file. Otherwise, changes in the queue will not activate the correct server program.

To assign the mailbox database trait to a queue, do the following:

- a. Open the Queue Properties dialog box for a queue, such as `SYSTEM.SEK`. See "Changing Database Traits" on page 5-21 for more information.

- b. Click on the Maintain tab. See “Maintain Tab” on page 5-38 for more information.
- c. Select the Standard radio button.
- d. Type in the mailbox number, such as 260.
- e. Click OK to save changes.

More detailed information for steps 6-9 is available in “Adding a Server Program to the System” on page 14-3.

6. (Optional) Test your configuration changes for the Avstar Server on which you added the seek server program.
7. Reconfigure the system.
8. Use the `restart` console command to start the seek server.
9. (Optional) Back up site files using the `sitedump` command.

## Fast Text Search (FTS) Servers

Fast Text Search (FTS) is an alternative to the seek feature. FTS allows fast, efficient searching through large quantities of text material using indexes.



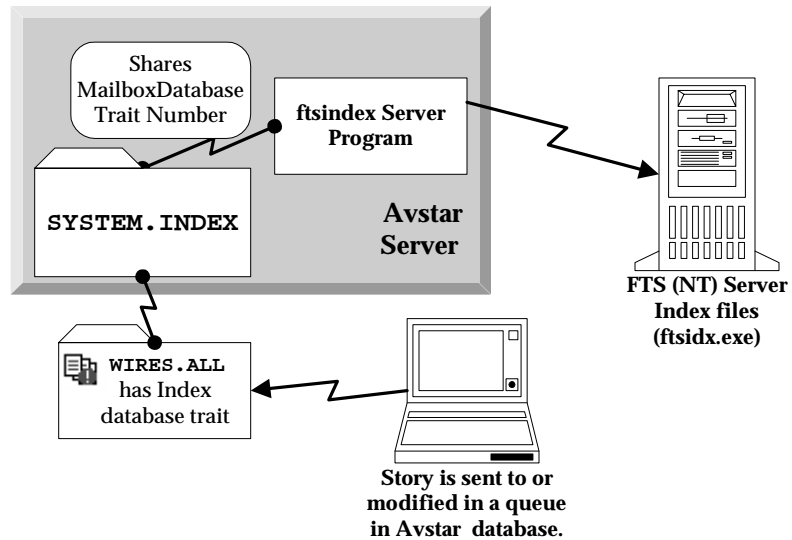
*While both seek and FTS searches operate in the background—that is, the user can continue with other activities at the Avstar Workstation while the system searches the Avstar database—FTS searches will generally be faster than seek searches.*

### FTS Workflow


Fast text searching involves two separate activities: indexing data and searching the index for data. Different components are required for these separate activities to function.

## FTS Indexing

Figure 14-6 shows the workflow Avstar NRCS follows to create Index files of database stories, which can be queried later.



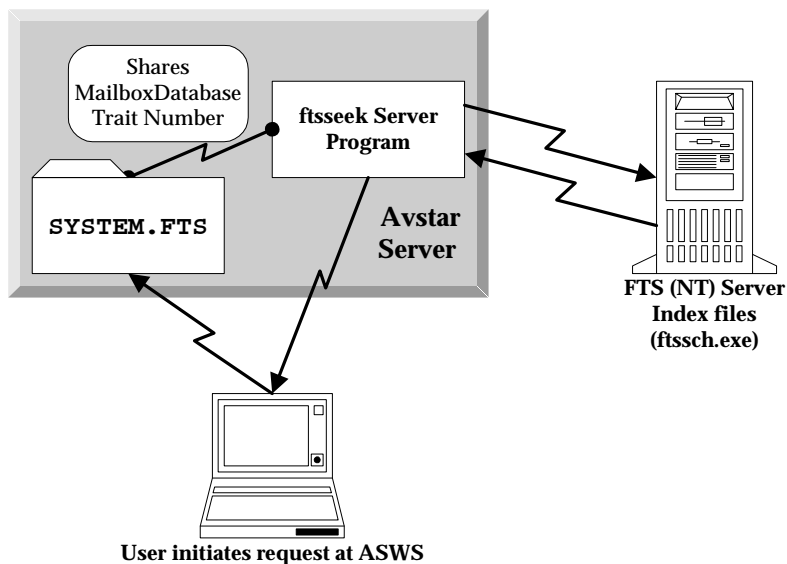
**Figure 14-6 Workflow for FTS Indexing**

 In a fast text search, the FTS server gets duplicate copies of material from those queues and folders in the Avstar database marked with the Index icon (as shown, left). The system administrator identifies which queues and folders the system should index. See “Batch Indexing” on page 14-75 for more information. The material in those queues and folders is fully indexed in the FTS server, so every word is added to an Index file.

For each word, the index lists information about all stories where that word was encountered. The index is updated each time a story is added, changed, or deleted in an indexed queue. For instance, in Figure 14-6, a story is added to **WIRES.ALL**. The system identifies that queue as an indexed queue and prompts the **ftsindex** server program to forward copies of the data to the FTS server for indexing.



Figure 14-7 shows the workflow for a user's FTS request—the procedure Avstar NRCS follows to search Index files and respond with results (pointers to database stories containing search criteria specified by the user).



**Figure 14-7 Workflow for FTS Query**

In the above example, the user sends an FTS query to the Avstar Server. The request is received in `SYSTEM.FTS`, which notifies the `ftsseek` server program to conduct the search. (This is accomplished because both the `SYSTEM.FTS` queue and the server program share the same mailbox number.) The `ftsseek` server program contacts the FTS server, which conducts a search of its Index files for data matching the user's search criteria, and then responds with pointers to those stories located in the Avstar database. The `ftsseek` server program forwards the FTS server's results to the user at the Avstar Workstation.

Fast text search acts much like an index in a book. In other words, the fast text search stores references to where the information you search for is stored. For instance, if you were to search for the word *Clinton*, the FTS server will note that the word appears in story 1, 8, 12, and 16.

If you looked for the word *Gore*, the FTS server will find the word in stories 2, 8, and 12. If you look for stories containing *Clinton & Gore*, you can view stories that appear in both lists (in this example, stories 8 and 12). This is faster than examining every story in the queue to see which contain both words. Instead, the system does that for you, and you benefit from the search.

If you invoke a background search (Find All or Find Global) for material in a folder or queue marked with the Indexed icon, Avstar sends your search request to the FTS server. The FTS server will look up the words you specified in the index, and return a list of the resulting stories to the Avstar Workstation.

Not all queues can be searched with the fast text search, simply because it takes space and effort to maintain an index. Therefore, system administrators will usually index only those queues searched often. You can determine which queues are available for fast text searches in three ways:



- In the Directory panel, the queue will display an Index icon (as shown in the WIRES . ALL queue, left)
- In the Find All dialog box, you will see the word “Indexed” in the title
- In the Directory/Queue Properties dialog box, the directory/queue will have the index database trait

## Installing FTS Components

FTS is an optional feature that requires certain configurations within the Avstar system as well as additional hardware and setup. For instance, FTS requires the set up of two utility programs (ftsindex and ftsseek) on Avstar Servers. Additionally, two executable files (ftsidx.exe and ftssch.exe) must be installed and running on a Windows® NT server. These components are described in further detail in the next sections of this chapter.

## ...On the Windows NT Server

The Windows NT server running the two FTS programs (`ftsidx.exe` and `ftssch.exe`) is known as the *FTS server*. This computer is not to be confused with the two utility programs (discussed in “...On Avstar Servers (UNIX)” on page 14-69) known as `ftsindex` and `ftsseek` servers.

The FTS server should be running the Windows NT 4.0 operating system. For information on the current service pack, contact iNews Customer Support.

### Installing `ftsidx.exe` and `ftssch.exe`

To install the executable files (FTS programs) required for FTS, do the following:

1. Insert the Avstar NRCS CD into the FTS server's CD ROM drive.
2. Navigate to the FTS folder on the CD.
3. Double-click on the setup program (`setup.exe`) and follow instructions as they appear on screen.
4. The setup program will install several files, including `ftsidx.exe` and `ftssch.exe`. Make a note of the directory (folder) where they are installed.
5. A batch file—used to start the FTS processes (executable files installed in step 4)—is located on the installation CD in the Misc folder. The batch file is called `go-fts.bat`.
  - a. Copy that file to the directory in which you installed the executable files, such as:  
`C:\Program Files\Avstar Systems\FTS`
  - b. (Optional) Create a shortcut to the batch file on the desktop. (Right-click and drag the file to the desktop, then select Create Shortcut Here when the pop-up menu appears.)

This provides you with a desktop shortcut you can double-click to launch the FTS programs: `ftsidx.exe` and

`ftssch.exe`. See “Starting `ftsidx.exe` and `ftssch.exe`” on page 14-68 for more information.

- c. The batch file contains two command lines in the code that you may need to modify based on the ports used by FTS. To open the batch file for editing, right-click on the file and select Edit from the pop-up menu. Locate the two lines in the code that look like this:

```
@cgi-fcgi -start -connect :6100 ftsidx.exe
@cgi-fcgi -start -connect :6101 ftssch.exe
```

The `W_BINDFTSI` and `W_BINDFTSS` dictionaries are defined in the `/site/dict/words` directory. See Table 14-2 on page 14-70 for more information.

The number 6100 corresponds to the Index `<port>` defined for `W_BINDFTSI` on the Avstar Server, and 6101 corresponds to the Search `<port>` defined for `W_BINDFTSS` on the Avstar Server. Ports 6100 and 6101 are defaults used by the FTS Index and Search programs.

6. You must also create a directory (folder) on the FTS server that will be used by the Index program to hold index files it makes. You can create the directory using Windows NT Explorer.
  - a. Click Start.
  - b. Locate and select Windows NT Explorer in the Start menu.
  - c. Navigate to where you want the new directory to be and right-click.
  - d. Select New Folder from the pop-up menu and name it, such as Index. For instance, the full pathname could be: `C:/Index`.

### Starting `ftsidx.exe` and `ftssch.exe`

The executable file, `ftsidx.exe`, is known as the Index program and the executable file, `ftssch.exe`, is called the Search program. These programs are started from the MSDOS command console window.



**The MSDOS command console window must not be closed (it may be minimized) after starting the `ftsidx.exe` and `ftssch.exe` programs. Closing the window will cause the two programs to exit.**

A batch file, called `go-fts.bat`, provided on the installation CD, can be used to launch the Index and Search programs in the MSDOS command console window. Otherwise, the programs can be individually launched manually.

To manually start the search and index programs on the NT system, do the following:

1. Open the MSDOS command console window.
2. Enter the following commands:

```
cgi-fcgi -start -connect :6100 ftsidx.exe  
cgi-fcgi -start -connect :6101 ftssch.exe
```

The 6100 corresponds to the index `<port>` defined for `W_BINDFTSI` on the Avstar Server. The 6101 corresponds to the search `<port>` defined for `W_BINDFTSS` on the Avstar Server.

The `W_BINDFTSI` and `W_BINDFTSS` dictionaries are defined in the `/site/dict/words` directory. See Table 14-2 on page 14-70 for more information.



*If you restart FTS programs, you must also restart `ftsseek` and `ftsindex` server programs on the Avstar Server-side. Start NT-side FTS programs first; otherwise, `ftsseek` and `ftsindex` will exit with an error message on the Avstar console.*

### Stopping `ftsidx.exe` and `ftssch.exe`

Use the Windows NT Task Manager to stop (kill) the FTS programs, `ftsidx.exe` and `ftssch.exe`, as follows:

- a. Open the NT Task Manager.
- b. Click the Processes tab.
- c. Select `ftsidx.exe`
- d. Click EndTask, then OK in the confirmation dialog.
- e. Repeat steps c-d, selecting `ftssch.exe` instead.

### ...On Avstar Servers (UNIX)

There are two utility programs used by FTS known as `ftsindex` and `ftsseek` servers. These must be set up on Avstar Servers—that is, comput-

ers running the UNIX operation system and Avstar Server application, such as server A and server B.

To set up these utility programs, you may need to create some queues, such as `SYSTEM.INDEX` and `SYSTEM.FTS`, if they do not already exist in the Avstar database. Also, some editing of files in the Site directory is required, particularly files in the `/site/dict` directory. To edit files in `/site/dict`, you will use `ed`, the UNIX line editor. See “Using the UNIX Line Editor” on page 10-4 for more information.

Table 14-2 shows dictionaries defined in the Queues and Words directories in `/site/dict`. If the dictionary name starts with a `Q`, the dictionary is in `/site/dict/queues`. If it begins with a `W`, it is defined in `/site/dict/words`.

**Table 14-2 Dictionaries for FTS in `/site/dict`**

<b>FTS Dictionaries</b>	<b>Explanation</b>
<code>Q_INDEX</code> queue	Identifies the queue—typically <code>/SYSTEM.INDEX</code> —that holds index requests from various server programs. The <code>ftsindex</code> program reads index requests from this queue, which is not visible on Avstar Workstations (ASWS) and will always show up “empty” on VT sessions.
<code>Q_FTS</code> queue	Identifies the queue—typically <code>/SYSTEM.FTS</code> —that holds indexed search requests, much like <code>SYSTEM.SEEK</code> holds seek requests.
<code>W_INDEXED</code>	The word— typically set to <code>/INDEXED</code> — used in indexed search requests to indicate that an indexed search is being requested. This type of search is similar to “fast” and “slow” seek searches.

**Table 14-2 Dictionaries for FTS in /site/dict**

FTS Dictionaries	Explanation
W_INDEXBASE	Defines location and name of index files on the FTS server. Its format is <pathname>/<basename>. The <pathname> is a directory that exists on the FTS server. The <basename> is any name you choose for index files. The FTS server will create four index files with .inX appended to the <basename>, where X is 1, 2, 3, or 4, such as verdi.in1, verdi.in2, and so forth. For instance, W_INDEXBASE could be defined in the following two ways: //index/verdi or /c:/index/verdi
W_HITSWEIGHT	Weight parameter that determines search results ranking by how many times a word appears in a story (number of hits). Value ranges from 0-100. Default is 0.
W_AGEWEIGHT	Weight parameter that determines search results ranking by age of stories (newer stories have higher weight and appear at top) appears in a story. Value ranges from 0-100. Default is 100.
W_LOCWEIGHT	Weight parameter that determines search results ranking by location of words searched for (words or "hits" closer to start of story have higher weight) appears in a story. Value ranges from 0-100. Default is 0.
W_BINDFTSI	Identifies the binding address for the FTS server and its port, used for indexing. The format is /<server>:<port>. The <server> is the host-name of the FTS (NT) server and <port> is the dedicated port number used for enhanced search indexing. The default is /dexy:6100 where dexy is the server name and 6100 is the TCP port.

**Table 14-2 Dictionaries for FTS in /site/dict**

FTS Dictionaries	Explanation
W_BINDFTSS	Identifies the binding address for the FTS server and its port, used for searching. The format is /<server>:<port>. The <server> is the host-name of the FTS (NT) server and <port> is the dedicated port number used for enhanced text searching. The default is /dexy:6101 where dexy is the server name and 6101 is the TCP port.

### Setting up ftsseek and ftsindex Servers

To set up the ftsseek and ftsindex servers, do the following:

1. At an Avstar Workstation, verify whether the Index and FTS queues exist in the System directory. If `SYSTEM.INDEX` and `SYSTEM.FTS` do not exist, you must create them.
2. At the Avstar console, add a definition for `Q_INDEX` to `/site/dict/queues`. For instance:

```
Q_INDEX /system.index
```
3. Add a definition for `Q_FTS` to `/site/dict/queues`. For instance:

```
Q_FTS /system.fts
```
4. Configure ftsindex and ftsseek server programs.
  - a. Choose device and mailbox numbers for each server program.
  - b. Edit `site/config`, ensuring there is a server line for each server program. For instance:

```
server 261 ftsseek 261 - ;fts searches
server 262 ftsindex 262 - ;fts indexing
```
  - c. Assign mailbox numbers to the appropriate queues in the System directory. For instance, assign 262 to the queue defined for `Q_INDEX`, such as: `SYSTEM.INDEX`, and assign 261 to the queue defined for `Q_FTS`, such as: `SYSTEM.FTS`. Further

See "Adding a Server Program to the System" on page 14-3 for more information.



details for assigning mailbox numbers (database traits) to queues are found in “Assigning a Mailbox to a Queue” on page 14-17. Also, see “Changing Database Traits” on page 5-21 for more information.

- d. Assign a queue form and story form to queues defined for `Q_INDEX` and `Q_FTS` (`SYSTEM.INDEX` and `SYSTEM.FTS`, respectively). The forms should be the same queue and story forms assigned to the queue defined for `Q_SEEK`, such as `SYSTEM.SEEK`. See “Assigning a Form as a Queue or Story Form” on page 8-11 for more information.
5. Add a definition for `W_INDEXED` to `/site/dicts/words`. It should be set to `INDEXED`. For instance:
 

```
W_INDEXED    /INDEXED
```
6. Add a definition for `W_INDEXBASE` to `/site/dicts/words`. Ensure the associated path exists and has appropriate access privileges. It should be an absolute path. For instance:
 

```
W_INDEXBASE  //index/verdi
```

-OR-

```
W_INDEXBASE  /c:/index/verdi
```
7. Add weight definitions for search results ranking factors (age, number of hits, and location of hits) in `/site/dict/words`.



*If all weight definition parameters (`W_HITSWEIGHT`, `W_AGEWEIGHT`, and `W_LOCWEIGHT`) are set to zero, ranking order is undefined. Any setting that is beyond the range of 0-100 is rounded to within the correct range. See Table 14-2 on page 14-70 for more information.*

8. Set the binding address for indexing by adding a definition for `W_BINDFTSI` to `/site/dict/words`. Use the format `/<server>:<port>`. For instance:
 

```
W_BINDFTSI   /dexy:6100
```

9. Set the binding address for searching by adding a definition for `W_BINDFTSS` to `/site/dict/words`. Use the format `/<server>:<port>`. For instance:

```
W_BINDFTSS    /dexy:6101
```



*You must define separate port numbers for `W_BINDFTSI` (indexing) and `W_BINDFTSS` (searching) on the FTS server. See Table 14-2 on page 14-70 for more information.*

### Starting and Stopping `ftsindex` and `ftsseek` on the Avstar Server

To start both FTS server programs (located on the Avstar Server), use the `restart` console command in the following format:

```
restart <device number> <device number>
```

For instance, select the appropriate server at the Avstar console, and type:

```
restart 261 262
```

To stop both FTS server programs (located on the Avstar Server), use the `stop` console command in the following format:

```
stop <device number>
```

For instance, select the appropriate server at the Avstar console, and type:

```
stop 261
```

```
stop 262
```



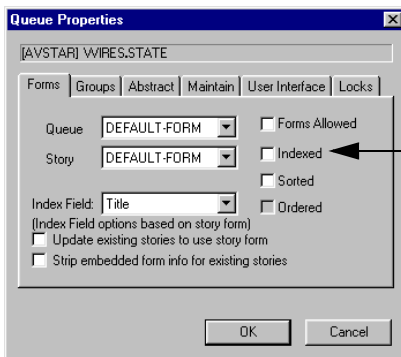
*Stop both `ftsseek` and `ftsindex` programs before the FTS programs on the FTS (NT) server. This will help ensure a stable, quiescent index base for future use when you restart enhanced search.*

## Batch Indexing

Index base is the set of four Index files on the FTS (NT) server, as defined in `W_INDEXBASE`, in `/site/dict/words`. See Table 14-2 on page 14-70 for more informaton.

Batch indexing is a procedure by which a system administrator can define queues to be indexed or de-indexed, and cause an update of the index base to incorporate the current state of the queues' stories in the index base. Batch indexing can be performed at either the queue or directory level by applying the index database trait. See "Changing Database Traits" on page 5-21 and "Database Traits Summary" on page 5-25 for more information.

To mark a queue as indexed and schedule an index base update, apply the Index database trait to the queue using the Queue Properties dialog box. This command marks the queue, `WIRES.STATE`, as indexed and submits a request to update entries in the index base for all stories in the queue. The queue will appear with the Indexed icon (shown at left) in the Directory panel of the Avstar Workspace.



Select Indexed check box to apply the index database trait. Click OK to save changes.

Similarly, to mark a queue as deindexed and schedule an index base update, uncheck the Indexed check box in the Queue Properties dialog box. This command marks the queue, `WIRES.STATE`, as not indexed and submits requests to remove all story entries for the queue from the index base.

### Batch Indexing Directories (Folders)

To mark all queues at every level under a directory (folder) as indexed and schedule an index base update, apply the index database trait to the directory, using the Directory Properties dialog box. Similarly, to mark all queues at all levels under a directory as deindexed and schedule uncheck the Indexed check box in the Directory Properties dialog box.



*The database traits program will ignore requests to index or deindex the queues defined for the dictionary entries Q\_SEEK, Q\_FTS, Q\_INDEX, and Q\_DEAD in /site/dict/queues.*

### Dynamic Indexing

Dynamic indexing is a function of the Avstar Server that automatically triggers an index change when a story is added, changed, moved, or deleted. Set index database traits for queues to be indexed as in the Batch Indexing instructions, and the Avstar Server will automatically schedule updates to the index base whenever a story in an indexed queue changes.

## Archival and Backup

### Copying and Archiving the Index Base Files

To save the current state of an index base, you must copy the Index files to another location and then archive/back up the files as you would normally do for an NT system. (See Windows NT Server documentation for details on backing up and archiving files.)



*Archived index bases may be out of sync with the active Avstar database.*

## Removing the Index Base and Reindexing (Optional)

If an index base becomes unusable, you should remove the index base on the FTS (NT) server and reindex, by doing the following:

1. Stop the `ftsseek` and `ftsindex` server programs. See “Starting and Stopping `ftsindex` and `ftsseek` on the Avstar Server” on page 14-74 for more information.
2. Stop the `ftssch.exe` and `ftsidx.exe` programs on the FTS (NT) server. See “Stopping `ftsidx.exe` and `ftssch.exe`” on page 14-69 for more information.
3. Open the folder—typically called `INDEX`—where the Index files reside.

This is the `<path>` portion of the `//<path>/<basename>` defined for `W_INDEXBASE` in `/site/dict/words` on the Avstar Server. For instance, `W_INDEXBASE` is defined as follows:

```
W_INDEXBASE      //index/verdi
```

In this example, the folder would be `index` on the FTS server.

4. Select all files whose names match the `<basename>` portion of the `//<path>/<basename>` defined by `W_INDEXBASE`.

Continuing with the example in step 3, the `<basename>` is `verdi`, so you would select all files matching `verdi`, such as `verdi.in1`, `verdi.in2`, and so forth.

5. Delete the selected files.
6. Restart the `ftssch.exe` and `ftsidx.exe` programs on the FTS (NT) server. See “Starting `ftsidx.exe` and `ftssch.exe`” on page 14-68 for more information.
7. Restart `ftsseek` and `ftsindex` server programs on the Avstar Server. See “Starting and Stopping `ftsindex` and `ftsseek` on the Avstar Server” on page 14-74 for more information.
8. At the Avstar console, issue a `dbtraits` command with the `reindex` option for one or more folders that may contain indexed queues. This will traverse the folder's hierarchy and schedule index requests for each indexed queue found.

For instance, type:

```
dbtraits wires reindex
```

This will cause the system to revisit all queues anywhere under the Wires directory and schedule indexed queues for indexing.

## Print Servers

A print server allows users to employ the `print` command to print stories to any queue in your system. When the story reaches the queue, it is formatted so that it appears on the screen exactly as if you had sent it to a system printer. This is referred to as printing to a story.



*While configuring a print server allows Video Terminal (VT) users to employ `print` when printing to local printers, print servers use the system printing function, not the local printing function.*

Not every newsroom wants to use these features; only those that request it have a print server added to their system during installation. If you want the server, install it as described in “Adding a Print Server” on page 14-79.

Because a print server’s only job is to print stories to queues (and handle local printer requests for VTs), it does not need a mailbox or a job list. Installing one requires that you choose a device number for the print server, add the print server to your system’s configuration file, and assign it a profile.

You need install only one print server to make the print-to-story and local printing features available to everyone. In fact, you can only install one such server on your system. Only one user at a time can use the print-to-story (or VT local printing features). Other users who submit print requests while the server is busy have their requests queued, to be sent when the server is available again.

## Adding a Print Server

Installing a print server (utility program) requires only that you choose a device number for the print server, add the print server to your system's configuration file, and assign it a profile. A print server does not need a mailbox or a job list.

To add a print server to your system, do the following:

1. Choose a device number for the print server.
2. Add the server program information to the configuration file (`/site/config`). Changing this file requires the use of `ed`, the UNIX line editor. See "ed, the Line Editor" on page 10-1 and "Changing the Configuration File" on page 11-15 for more information.

- a. Select *all* Avstar Servers. See "Selecting Servers" on page 2-5 for more information.

- b. Use `ed` to open the configuration file for editing by typing:

```
AVSTAR-A: ed /site/config
editing /site/config
1259
```

- c. The print server's device number must appear in a `servers` line in the host definition for the Avstar Server on which it runs, such as server B. Move to the `servers` line in server B's host definition. It should appear something like this:

```
servers 257 258 259 260 261 262
```

Then, add the print server's device number, such as 263, to the `servers` line in server B's host definition.

```
servers 257 258 259 260 261 262 263
```



*In addition to adding the print server to the standard host definition for the Avstar Server on which you want it to run, make sure that you also add it to each alternate host definition.*

3. Create a configuration line for the print server. A print server configuration line uses the standard server format shown here:

```
server <device #> printserver - -
```

The new line should look like this:

```
server 263 printserver - - ;print server
```

Do *not* use an uppercase (W) in step 4. See “ed, the Line Editor” on page 10-1 for more information.

4. After adding the line, type:

```
w
```

```
1565
```

```
q
```

5. Create a printer profile exactly as you would for a printer connected to a PCU, as described in Chapter 12. The print server’s profile must be in a file named after the print server’s device number. For instance, put it in: `/site/printers/263`.
6. (Optional) Test your configuration changes for the Avstar Server on which you added the seek server program.
7. Reconfigure the system. You do not need to stop anything to reconfigure the system.
8. Use the `restart` console command to start the print server.
9. (Optional) Back up site files using the `sitedump` command.

The style options you include in this profile are also available to VT users printing stories locally.

More detailed information for steps 6-9 is available in “Adding a Server Program to the System” on page 14-3.

## Mail Servers

Your system provides a mail feature that allows users to send mail messages to other users. To enable the mail feature, a utility program known as the mail server must be installed on your system. Generally, this is done when your system is installed.



## Disabling Mail to All Users

Ordinarily, a user can send mail to `all`, which sends a copy of the mail story to every user on the system. To disable this feature, include an alias for `all` in a story in `SYSTEM.GROUPS` that maps the word `all` onto a nonexistent user name. The new alias may look like this:

```
alias all
not-allowed
```

Because `not-allowed` is not a valid user on your system, anyone who tries to send mail to `all` will get the error message:

```
Returned mail: User unknown.
```

## Using Network Mail

Your system has a hosts file that it uses to communicate with other devices on the network. This file is found in `/etc/hosts`. Each Avstar Server's hosts file is like a directory that includes the network addresses of all computers on the network. Avstar Servers use that directory to find the address of other computers. See "The Hosts File (`/etc/hosts`)" on page 11-17 for more information.

The typical format of a line in the hosts file lists the Internet address of a computer or other network device, followed by its name in lowercase and then in uppercase.

For instance, here is an `/etc/hosts` file for a three-server system:

```
124.4.0.1 avstar-a AVSTAR-A a A
124.4.0.2 avstar-b AVSTAR-B b B
124.4.0.3 avstar-c AVSTAR-C c C
```

If you edit the host file, run `grpcheck` to have your changes take effect.

```
AVSTAR-A# grpcheck -v system.groups
```

## Using 8-Bit Characters in Mail

You can use 8-bit characters, such as accented letters (ñ, ó, and á), in both mail messages and mail addresses. You send mail in a format compatible with the MIME standard. By default, 8-bit characters will be represented in the MIME standard `quoted-printable` format. See RFC 2045 at <http://www.rfc-editor.org/rfc/rfc2045.txt> for more information. You can change this behavior by setting the `EIGHTMAIL` environment variable in `/etc/.profile`.

`EIGHTMAIL=x_underscore-prefix` causes mail to be encoded using the Avstar system underscore format. `EIGHTMAIL=8bit` or `EIGHTMAIL=` turns off 8-bit encoding and sends unencoded 8-bit characters.

Only two programs use `EIGHTMAIL`, namely `grpcheck` and the mail server.

`Grpcheck` normally converts 8-bit characters in user, group, and alias names to our proprietary `underscore-prefix` format when building the mail alias file (`/site/.mail/aliases`). This occurs when `EIGHTMAIL` is not found in `grpcheck`'s environment. If `EIGHTMAIL=` or `EIGHTMAIL=8bit` is present in `grpcheck`'s environment, conversion is not done (the 8-bit characters are put into the alias file). If `EIGHTMAIL` is in `grpcheck`'s environment but is set to some other value, it is ignored and `grpcheck` behaves as when `EIGHTMAIL` is not found in the environment.

The mail server normally converts 8-bit characters in the From, To, and CC names to `underscore-prefix` format and converts 8-bit characters in text to `quoted-printable` format for outgoing mail messages. This occurs when `EIGHTMAIL` is not found in mail server's environment or when `EIGHTMAIL=quoted-printable` is found in mail server's environment. If `EIGHTMAIL=` or `EIGHTMAIL=8bit` is present in mail server's environment, no conversion is done (the 8-bit characters are included in the outgoing mail message). If `EIGHTMAIL=x-underscore-prefix`, mail server will convert 8-bit characters in the From, To, and CC names and 8-bit characters in text

to underscore-prefix format. The mail server will include a Content-Transfer-Encoding: <type> statement in the outgoing mail message. The <type> will be set to one of quoted-printable, x-underscore-prefix, or 8bit. This will be used by the mail receiving program to determine how to convert back into 8-bit characters.

The mail server's behavior may be more easily understood, as shown in Table 14-3.

**Table 14-3 Mail Server's Behavior**

Environment	From/To/CC Names	Text
(nothing)	underscore-prefix	quoted-printable
EIGHTMAIL=quoted-printable	underscore-prefix	quoted-printable
EIGHTMAIL=x-underscore-prefix	underscore-prefix	underscore-prefix
EIGHTMAIL=	(no conversion)	(no conversion)
EIGHTMAIL=8bit	(no conversion)	(no conversion)



*All environment variables put into /etc/.profile should be placed before the line exec sosh.*

## Character Conversion Table for Underscore-Prefix Format

The Avstar system uses the Character Conversion table (see Table 14-4) to convert 8-bit characters used in mail stories into 7-bit representation strings when mail is sent and then converted back into their 8-bit values when the mail is received at another Avstar system.

If you are sending mail from a UNIX system to Avstar NRCS, the Mail Compose strings in column four are required only in the address of the mail; you can use either these strings or the standard

VT220 compose sequences within the message. Use the corresponding lowercase characters in the address.



*Character conversion data in Table 14-4 originates from the MS (Windows) 1252 code page, which is used for such languages as: English, German, French, Icelandic, Norwegian, Italian, Portuguese, Spanish, Dutch, Finnish, and Swedish. Other languages, such as Arabic, Czech, Turkish, Polish, Hungarian, Slovenian, and so forth, use different code pages. Contact iNews Customer Support for assistance with these language conversions.*

**Table 14-4 Character Conversion Table**

<b>8-Bit Character</b>	<b>Decimal</b>	<b>Hex</b>	<b>Mail Compose</b>	<b>7-Bit String</b>
i	161	A1	_}}	_}}
ç	162	A2	_c-	_ c
£	163	A3	_l-	_l=
¥	165	A5	_y-	_y=
§	167	A7	_so	_}s
¸	168	A8	_xo	_xo
©	169	A9	_oc	_oc
ª	170	AA	_}a	_a_
«	171	AB	_~~	_~~
°	176	B0	_o&	_o&
±	177	B1	_+ -	_+ -
²	178	B2	_2+	_ ^2
³	179	B3	_3+	_ ^3

**Table 14-4 Character Conversion Table (Continued)**

<b>8-Bit Character</b>	<b>Decimal</b>	<b>Hex</b>	<b>Mail Compose</b>	<b>7-Bit String</b>
μ	181	B5	_u-	_u/
¶	182	B6	_}p	_}p
·	183	B7	_+*	_^.
1	185	B9	_1+	_^1
°	186	BA	_}o	_o_
»	187	BB	_``	_``
1/4	188	BC	_41	_41
1/2	189	BD	_21	_21
¿	191	BF	_{{	_??
À	192	C0	_`A	_`A
Á	193	C1	_A'	_A'
Â	194	C2	_A+	_^A
Ã	195	C3	_~A	_~A
Ä	196	C4	_A\$	_A\$
Å	197	C5	_A*	_A*
Æ	198	C6	_EA	_EA
Ç	199	C7	_}C	_C;
È	200	C8	_`E	_`E
É	201	C9	_E'	_E'

**Table 14-4 Character Conversion Table (Continued)**

<b>8-Bit Character</b>	<b>Decimal</b>	<b>Hex</b>	<b>Mail Compose</b>	<b>7-Bit String</b>
Ê	202	CA	_E+	_ ^E
Ë	203	CB	_E\$	_E\$
Ì	204	CC	_I	_I
Í	205	CD	_I'	_I'
Î	206	CE	_I+	_ ^I
Ï	207	CF	_I\$	_I\$
Ñ	209	D1	_~N	_~N
Ò	210	D2	_`O	_`O
Ó	211	D3	_O'	_O'
Ô	212	D4	_O+	_ ^O
Õ	213	D5	_~O	_~O
Ö	214	D6	_O\$	_O\$
Œ	215	D7	_OE	_OE
Ø	216	D8	_O-	_O/
Ù	217	D9	_`U	_`U
Ú	218	DA	_U'	_U'
Û	219	DB	_U+	_ ^U
Ü	220	DC	_U\$	_U\$
ÿ	221	DD	_Y\$	_Y\$

**Table 14-4 Character Conversion Table (Continued)**

<b>8-Bit Character</b>	<b>Decimal</b>	<b>Hex</b>	<b>Mail Compose</b>	<b>7-Bit String</b>
ß	223	DF	_ss	_ss
à	224	E0	_a`	_a`
á	225	E1	_a'	_a'
â	226	E2	_a+	_a^
ã	227	E3	_~a	_~a
ä	228	E4	_a\$	_a\$
å	229	E5	_a*	_a*
æ	230	E6	_ea	_ea
ç	231	E7	_}c	_c;
è	232	E8	_e`	_e`
é	233	E9	_e'	_e'
ê	234	EA	_e+	_e^
ë	235	EB	_e\$	_e\$
ì	236	EC	_i`	_i`
í	237	ED	_i'	_i'
î	238	EE	_i+	_i^
ï	239	EF	_i\$	_i\$
ñ	241	F1	_~n	_~n
ò	242	F2	_o`	_o`

**Table 14-4 Character Conversion Table (Continued)**

<b>8-Bit Character</b>	<b>Decimal</b>	<b>Hex</b>	<b>Mail Compose</b>	<b>7-Bit String</b>
ó	243	F3	_o'	_o'
ô	244	F4	_o+	_o^
õ	245	F5	_~o	_~o
ö	246	F6	_oS	_oS
œ	247	F7	_oe	_oe
ø	248	F8	_o-	_o/
ù	249	F9	_u`	_u`
ú	250	FA	_u'	_u'
û	251	FB	_u+	_u^
ü	252	FC	_u\$	_u\$
ÿ	253	FD	_y\$	_y\$

## Networking Two or More Servers Using Rx/Tx Links

This section introduces you to Rx/Tx links and explains their operation. Use Rx/Tx links to transfer stories between your Avstar system and other systems using the Ethernet network. To send stories across a network connection, use network Rx/Tx links. You can connect several servers or systems into a network using Rx/Tx links.

For one system to send stories to another, it must have a Tx (transmit) link configured. The other system must have an Rx (receive) link configured to receive stories the Tx link sends.

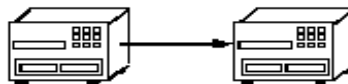




Information relevant to this section is also contained in “Assigning Field Validation” on page 14-26.

For instance, suppose you have an archive system in your newsroom. You could put a Tx link on your Avstar system that automatically sends any stories added to or edited in a particular queue to the archive system. Then add an Rx link to the archive system so it can receive those stories.

To send stories, the transmitting system must have a Tx link.



To receive stories, the receiving system must have an Rx link.

**Figure 14-8 Tx/Rx Links**

## Sending Story Forms

Your system can transmit story form information by sending the form name with a story or by sending the full form with a story. The first way reduces the time it takes the system to send a story to another system by omitting the form and sending only the text of the story, including any text contained in the form’s fields, and the number representing the form. When the receiving system gets a story without a full form, it uses the form number to assign one of its own forms.

If the receiving system gives a story a new form, it fills in the fields of that form as much as possible with the story’s original field text. If the original story had fields that are not present in the new form, the text from those fields is included in the main body text of the new story. This also occurs when the system transfers stories to an older release system that does not support new field types.

To ensure that transmitted stories have the same form on both systems, instruct the system to send the full form with the transmitted stories. This is done by adding a `sendform` line to the Tx job list. This

instruction must appear in the job list immediately following the scan line.

## Understanding the READY Field

The READY field usually contains text on the status of the story, such as HOLD, LOCK, WIRE, or READY. This information is transmitted with the story.

## Setting Automatic Update

When a new version of a story is transmitted to a queue that already contains an older version of the same story, you usually want the receiving system to replace the older version of the story with the new one. To do this, enable the `update` trait on the queue where the received stories are placed.

For instance, suppose an affiliate in Washington sends stories it uses for its six o'clock show to your system as these stories are added to or modified in the show's rundown queue. On receipt of these stories from Washington, your system puts them in `WASHINGTON.RUNDOWN.6PM`. You want your system to dispose of old versions of stories as new versions come in from Washington.

To set an automatic update, do the following:

1. Log in as a system administrator at an Avstar Workstation.
2. Navigate to the queue `WASHINGTON.RUNDOWN.6PM` and right-click on it.
3. Select Properties from the pop-up menu.
4. Click on the Maintain tab in the Queue Properties dialog box.
5. Click on the Update check box to turn on the update database trait in the queue `WASHINGTON.RUNDOWN.6PM`.
6. Save changes and close the dialog box.

See "Maintain Tab" on page 5-38 for more information.



If you change the update trait and stories are in the queue, those stories will not be replaced by updated versions. Turn on the trait when the queue is empty.

After you apply the update database trait, your system starts replacing older versions of stories in WASHINGTON . RUNDOWN . 6PM with new ones as they come in from Washington.

If you have a Tx link that sends stories to an update queue, any change you make to a story in a queue watched by the Tx link affects the copy of the story in the other system. This includes killing a story, which causes the Tx link to notify the receiving system that the story has been killed. The second system then kills its copy of the story as well. You can use the `ignore` or `delete` command to stop killed stories from being deleted.

## Updating Queue Considerations

A story's final destination queue and any intermediate queues through which the story passes before reaching the destination queue must have their update trait turned on. If not, the final destination queue does not get updated. It does not matter if the queue where the story was created has its update trait turned on.

As an example, Figure 14-9 shows which queues must have their update trait turned on if a story is transmitted directly from the queue in which it is created to its final destination queue.

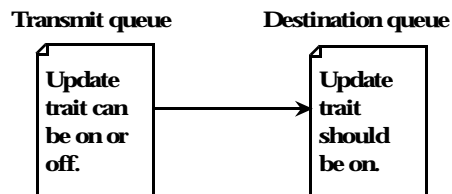


Figure 14-9 Story Queues

If the story passes through an intermediate queue before reaching its destination queue, both the intermediate queue and destination queue must have their update traits turned on, as shown in Figure 14-10.

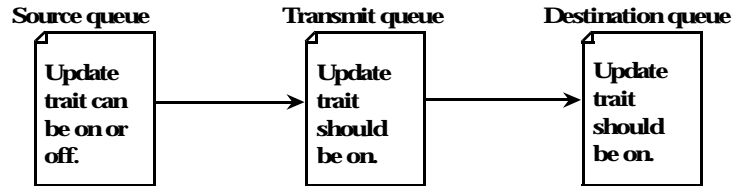


Figure 14-10 Intermediate Queues

Similarly, if a story received from one system is copied to another queue for transmission to a third system, enable the update trait on every queue through which the story passes, as shown in Figure 14-11.

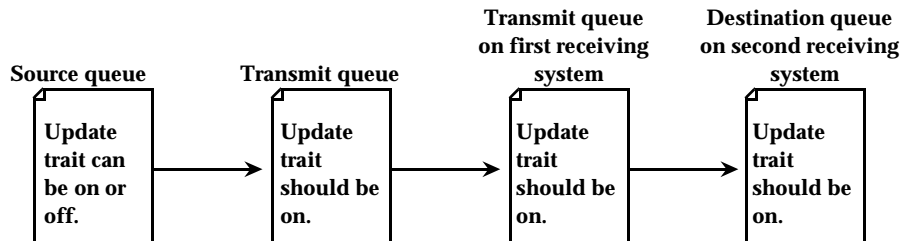


Figure 14-11 Transmitting to a Third System

## Changing Queue Order

To have the system transmit queue ordering information, add an `order` line to your network Tx job list. When a local queue is ordered, information is sent to the remote queue. That queue is ordered, identical to the local queue. This also applies to action servers.

Ordinarily, when a queue in the transmitting system is ordered, changes are not evident in the receiving system's corresponding queue. This is adequate for many queues, because their story order is not important. However, story order is very important for other

queues (for example, rundown queues). These queues must be identical on the local and remote systems.

Nothing must be done to the network Rx job list, but the destination queue must have its update trait turned on.

Also, additions and deletions to a sorted queue on the local system do not cause the system to transmit queue order change information. Ensure that you turn on the sort attribute in the destination queue.

## Adding Rx/Tx Links

If your Avstar system is connected to the same network as another Avstar system, you can use network Rx/Tx links to allow one system to automatically send stories to the other. For each Rx or Tx link, create an entry in your system's configuration file that defines where the link runs and how it sets up.

For instance, use network Rx/Tx links to have a networked Avstar system periodically send stories ready for archival to an Archive system on the same network. Set these links up so that people can place stories ready for archival in `archive` queues on the Avstar system. Then, on Sunday morning, the Avstar system moves these stories to queues on the Archive system.

Give the Avstar system a Tx link that performs two tasks. At 3 A.M. every Sunday, the first task moves any stories added to `ARCHIVE . SCRIPTS` since the previous Sunday to `SYSTEM . TRANSFER . SCRIPTS` on the Archive system. Then, at 4 A.M. every Sunday morning, the second task moves any stories added to `ARCHIVE . PACKAGES` that week to `FROM . NEWS . PACKAGES` on the Archive system.

## Adding Network Links

To add network Rx and Tx links to your Avstar system, do the following:

1. Ensure both systems are in the hosts file.

For two systems to use network Rx/Tx links to exchange stories, each must have the name of the computer(s) used by the other system in its hosts file.

Use the `cat` command to check each system's hosts file and ensure that necessary servers are listed. For instance, check the Avstar system:

```
AVSTAR-A: cat /etc/hosts
```

A message similar to the following will appear:

```
125.0.0.1    AVSTAR-A
125.0.0.2    AVSTAR-B
125.0.0.3    ARCHIVE-C
...
```

The Archive system's server (ARCHIVE-C) is listed in this file. Now check the Archive system's hosts file for AVSTAR-A and AVSTAR-B.

If you find that either system's hosts file does not contain the names of the computers belonging to the system to which you want it to connect, call iNews Customer Support for help in modifying the file.

2. Choose device numbers for the links.

Reserve a range of numbers for network devices and always choose the next available number from within that range. In the examples in this manual, we have reserved numbers 301-399 for network devices.

To see which numbers are taken, check your system's configuration file. For this example, choose device number 301 for the net-

work Tx link. On the Archive system, we use device number 101 for our network Rx link.

3. Create the Tx link job list queue in the database. See “Creating a New Queue” on page 5-6 for more information.

This queue must be in `SYSTEM.CONFIGURE`. Its name must begin with a three-digit number that matches the Tx link device number.

For instance, create a queue named 301 to hold the job list. Follow the number with a hyphen and a descriptive name, such as: `-txnet-to-archive`. The full pathname of the queue would be `SYSTEM.CONFIGURE.301-txnet-to-archive`.



*Network Rx links do not require job lists.*

4. Open the newly created Tx link job list queue.
5. Create the Tx link job list story in the database.



*Like action servers, Tx links use mailbox and timed-interval tasks. Tasks that you want the Tx link to execute as soon as a story is modified in a queue must be mailbox tasks, while those that you want the Tx link to execute at a specific time must be timed-interval tasks. See “Defining Tasks for Servers” on page 14-7 for more information.*

In a network Tx link’s job list, you must always follow the scan line with an `open` line and a `put` line, both of which are used only in this type of job list.

The `open` line uses the `open` command in the following format to open a network connection to the receiving system:

```
open <COMPUTER NAME> <username> [<format>]
```

The computer name must match exactly with the name in the `/etc/hosts` file, including case.

The user name must have the same password on both systems.



To improve security of your Tx link, apply the `blacklist` user trait to the user name (account) you list here, which prevents anyone from logging in under that name. This has no effect on name use in a job list. See “User Traits Summary” on page 4-5 for more information.

You must follow the `open` line with a `put` line, which uses the `put` command in the following format to send stories to a queue on the receiving system:

```
put <target queue name>
```

If you do not name a queue, `put` uses the destination queue assigned to the user you specified in the `open` command.



You can have only one `put` command in each task. If you want stories you transmit to be distributed to more than one queue, set up an action or distribution server on the receiving system to do this. See “Adding an Action Server” on page 14-22 and “Adding a Distribution Server” on page 14-38 for more information.

In this example, the network Tx link’s job list requires two timed-interval tasks. At 3 A.M. each Sunday, the first task sends copies of everything added to `ARCHIVE . SCRIPTS` in the last week to `SYSTEM . TRANSFER . SCRIPTS` on the Archive system. Then, at 4 A.M. every Sunday the second task sends copies of everything added to or changed in `ARCHIVE . PACKAGES` to `FROM . NEWS . PACKAGES` on the Archive system.



To ensure forms are transmitted with stories, include a `sendform` line in the Tx job list. The `sendform` line must be added to the job list after the `open` command and before the `put` command.

If you want any queue order changes in your scan queue to be transmitted to your remote queue, include an `order` line in the Tx job list. Also, make sure the update database trait of the destination queue is turned on.

After the Tx link has completed the first tasks, you may want it to remove stories it sent from the scan queue. To have it do this, end both tasks with the `remove` command.





If you use either `move` or `remove`, it must be the last command in the task's command set.

The finished job list looks like the following:

```

title          presenter    writer
txnet to archive      user5000
┌───────────┴───────────┴───────────┐
│ AT 3:00                                     │
│ on sun                                     │
│   scan archive.scripts                    │
│     open ARCHIVE_C link                    │
│     fulltemplate                           │
│     put system.transfer.scripts            │
│     remove                                  │
│ AT 4:00                                     │
│ on sun                                     │
│   scan archive.packages                    │
│     open ARCHIVE_C link                    │
│     fulltemplate                           │
│     put from.news.packages                 │
│     remove                                  │
└───────────┴───────────┴───────────┘

```

You can use the following job list commands in a network Tx link job list. (These are the same commands used by action servers.)

<code>scan</code>	<code>number</code>
<code>bscan</code>	<code>remove</code>
<code>dup</code>	<code>send-del</code>
<code>move</code>	<code>ignore-del</code>
<code>replace</code>	<code>validate</code>
<code>order</code>	<code>sendform</code>



Usually, you assign a read group to the System directory that prevents most users from reading stories in that directory. If you do not want to restrict all of the System directory, consider restricting who can modify stories in `SYSTEM.CONFIGURE`.

If write access for the destination queue on the receiving system in the `put` command is restricted, the connection must be established with the appropriate security to send stories to this queue. The user name you specify in the task's `open` command must belong to the necessary group on the receiving system.

6. Add links to the configuration file on each Avstar Server in your system. Changing this file requires use of `ed`, the UNIX line editor. See “`ed`, the Line Editor” on page 10-1 and “Changing the Configuration File” on page 11-15 for more information.

- a. Select *all* Avstar Servers. See “Selecting Servers” on page 2-5 for more information.

- b. Use the `ed` command to open and edit the configuration file.

```
ARCHIVE-C: ed /site/config
editing /site/config
1356
```

- c. Add the network Tx link's device number to a resource list or `reslist` line in the appropriate host definitions. For instance, put the network Tx link on server A by adding a `reslist` line in its host definition beneath the `servers` line, which is the last line in the host configuration.

```
servers 256 257 258 259 260 261 262 263
reslist 301
```



*To keep the network Tx link running if one of the Avstar Servers fails, add this line to each alternate host definition before proceeding.*

7. Add a configuration line (also known as `server` lines) for the Tx link. Configuration lines are generally grouped in one place in your configuration file. Append the network Tx link's configuration line below the last `server` line using the format shown:

```
special <device #> 0 txnet <mailbox> <device name>
```

The finished line (in bold) for the Tx link should appear at the end of the configuration line grouping:

```
server 256 action 256 actphon ;action svr
server 257 distribution 257 devname1 ;dist server
server 258 parallel 258 devname2
server 259 keyword 259 key1 ;keyword server
server 260 seek 260 seek ;seek server
server 261 ftsseek 261 - ;fts searches
server 262 ftsindex 262 - ;fts indexing
server 263 print 263 - ;print server
special 301 0 txnet- ;tx link
```

8. Create a configuration line for the Rx link. Like the configuration line for the Tx link, place this line at the end of the file.

The general format for a network Rx link configuration line is similar to that of a network Tx link:

```
special <device #> 0 rxnet <mailbox> <device name>
```

Since the Rx link does not use a mailbox, place a hyphen in the mailbox parameter. The Rx link does not use the device name field, so place a hyphen in that position, too.

The finished line should look like this:

```
special 101 0 rxnet - -
```

Do *not* use an uppercase (w) in step 9. See “ed, the Line Editor” on page 10-1 for more information.

9. When you are done making changes to the `/site/config` file, save your changes and exit the UNIX line editor by typing:

```
w
1401
q
```

10. (Optional) Use the `configure` command to test your configuration changes. For instance, type:

```
configure /site/config ab a
```

11. Reconfigure the system.
  - a. Select the master computer (typically server A). See “Selecting Servers” on page 2-5 for more information.
  - b. Become superuser. See “Becoming a Console Superuser” on page 3-2 for more information.

c. Type:

```
AVSTAR-A# offline
AVSTAR-A# configure
```

d. Type:

```
AVSTAR-A# online
```

A message similar to the following appears:

```
A Fri Oct 6 00:19:01 2000 msg System being configured
```

- e. Press CTRL-D to exit from superuser.
  - f. Repeat this step on the Archive system to incorporate the network Rx link into its operation.
12. When you see the System being configured message, start the Tx link by typing **restart** followed by the Tx link device number. For instance:

```
AVSTAR-A: restart 301
```

When the system displays a `Hot-to-go` message, the Tx link has been successfully started.



*A network Rx link is not restarted after you add it to your system. The system starts the Rx link automatically as soon as it receives something from the transmitting system.*

13. (Optional) Back up site files using the `sitedump` command any-time you make significant changes.

# CHAPTER 15

## *Web Publishing*

The Avstar Web publishing feature allows users to publish news on the World Wide Web (WWW) by transmitting stories to other computers in Hypertext Markup Language (HTML) format. This chapter covers the Web publishing installation and process.

This chapter includes:

- Overview
- Setting Up Tx/net to Send HTML
- Creating an HTML Story Form
- Adding Story Entity References
- NSML to HTML Conversion
- Using Optional Format Strings
- A Sample HTML Story Form

## Overview

The configuration procedure for Avstar's Web publishing involves:

- Setting up Tx/net to send Hypertext Markup Language (HTML)
- Creating the HTML skeleton story form
- Putting the HTML story form in a queue

Once this setup is complete, Avstar's Web publishing involves three steps:

1. Select a story from the news database.
2. Merge pieces of the story with an HTML skeleton story form to recreate the story as an HTML document.
3. Use the Tx/net facility to put the HTML document onto another computer, known as the Web Publishing server.

You can then revise the HTML-formatted story, as necessary, to complete the publishing process.

## Setting Up Tx/net to Send HTML

To set up Tx/net to send HTML:

1. Establish Tx links. See "Networking Two or More Servers Using Rx/Tx Links" on page 14-88 for more information.

The links (txnet) transfer files between systems. You can send stories to non-news systems. For instance, you can take files that exist on the Avstar database and send them on a Tx/net, transmitting UNIX files to a UNIX computer, DOS files to a DOS computer, and so forth. The only requirement is that the receiving computer support File Transfer Protocol (FTP) so the Avstar Server can send files to it.

2. Get the name of the Web server to which you will send the stories and be sure it is listed in `/etc/hosts`.

3. Create a valid login on the Web server.
4. Create that same login on the Avstar Server.



*The Web server and Avstar Server must have matching logins with matching passwords.*

5. Create an entry in the configuration file for the Tx/net to specify sending files in HTML format rather than the standard Avstar format, News Story Markup Language (NSML).

Configuring files for HTML is the same as configuring for Tx/net to another computer. The HTML files are configured identically, except for the `open` command syntax, as shown:

```
open <computer name> <user name> <format> [<queue
name>] [<story name>]
```

- a. Specify the Web publishing server name (to send the stories) as the `computer name`.
- b. Specify the `user name` that gives you access to the Web publishing server (common user account).
- c. Specify the `format` as HTML to enable Web publishing.
- d. (Optional) Specify the `<queue name>` as the queue that contains the HTML skeleton story form. If no queue is specified, a default is used. See “Default HTML Skeleton Story Form and Queue” on page 15-4 for more information.



**If you specify a queue, the entire queue name must be specified in the `open` command. Otherwise, the system will issue a “Failed to get form from...” error message when Tx/net is restarted. Also, if you specify a queue that does not contain an HTML skeleton story form, Web publishing will not work.**

- e. (Optional) Rather than use the default, which is the first story in the queue, specify the `story name` in the queue.

Here is an example of a correctly written `open` command line:

```
open ntsrvr webtx html system.webforms.s.script-tx
```

6. After you have edited the txnet job list, restart txnet for your changes to take effect.

When HTML stories are placed on the Web server, they are assigned a randomly generated, unique filename, similar to the following:

```
f6daab8.html
f6daab9.html
f6daaba.html
```

Additionally, a story named `published.html` is created in the directory written to. The `published.html` file contains a listing of each file exported to the server, in that directory, referenced by the quick index field of the story. The format is:

```
<A HREF="qstamp.html">quickindex &nbsp;  </A><BR><CR>
<LF>
```

Here is an example:

```
<A HREF="f6daab8.html">pre-show tease &nbsp;  </A><BR>
<A HREF="f6daab9.html">bosnia &nbsp;  </A><BR>
<A HREF="f6daaba.html">fort bragg &nbsp;  </A><BR>
```

This story could be used as a base for creating an index page. Generation of the `published.html` file can be turned off by a job list command after the scan line: `publish no`. The default for this job list command is `yes`. See “Job List Commands” on page A-42 for more information.

## Default HTML Skeleton Story Form and Queue

The default HTML skeleton story form (template) is shown in the next chapter.

If you do not specify a queue in the configuration file where the HTML skeleton story form resides, the system uses the default queue. The default queue and default HTML skeleton story form are specified in the dictionary entries `Q_WEBPUB_FORMS` for the queue and `W_WEBPUB_FORM` for the story form. The dictionary entries indicate to



the system in which queue the default HTML skeleton story form resides. The default queue is `SYSTEM.WEBFORMS.P.PUBLISH-FORM`. If the queue does not exist, or does not have an HTML skeleton story form, a hard-coded default Web form is used instead.

## Creating an HTML Story Form

The HTML skeleton story form defines the appearance of stories transmitted out of Avstar NRCS—that is, exported in HTML format. The form is stored as a news database story whose text is in HTML, with optional embedded references to story entities.

The Avstar system contains an HTML skeleton story form to use as the default form, but you can customize it or create new forms. See “Default HTML Skeleton Story Form and Queue” on page 15-4 for more information.

## Adding Story Entity References

In creating the HTML skeleton story form for exporting, you use HTML tags and add story components or entities.

The three basic entities of a story are:

- **Field IDs** – Story information such as story title, creation date, writer, and so forth. It is the information you see in a Queue or Story Form panel.
- **Body** – Story text.
- **Production cue set** – Production cues entered in the Instruction pane—to the left of a story’s text.

Specify entities in the HTML form using an ampersand (&) to mark the beginning of an entity and a semicolon (;) to mark the end. When put in this notation, an entity is called a *story entity reference*.

The following section shows the format for each story entity and its corresponding story entity reference:

- Field ID format is:

```
&f-<field_id>[<optional format string>]
```

The field ID is replaced in the HTML output by information associated with it. For instance, to create a field ID such as the title of a story to appear in the HTML story form, use the following:

```
<title>&f-title;</title>
```

You are combining HTML tags with the story entity reference. The information in the field ID is now a title in the HTML form, and when it is merged with an actual story, it will contain the story's title, in HTML format. For instance, the title will look like this:

```
<title>LOTTERY-WIN</title>
```

The `optional format string` defines the format to be applied to the field ID entity. The string applies to fields that are time-related, such as the date or length of a story. Format strings are optional and defaults are assumed if they are not specified. See the section on optional format strings for more information on how to modify this default display for date or time fields.

- Body format is:

```
&body;
```

When encountered in the form, it is replaced by the story text in the resulting HTML output.

- Production cue format is:

```
&aeset;
```

When encountered in the form, it is replaced in the resulting HTML output by a list of production cues defined in the production cue set, a collection of all production cues for the story. Production cues—also known as anchored elements—are anchored at specific positions in the story text. How they display depends on their content and the application displaying them.

The term `aeset` stands for anchored element set.

The rules for entity reference names are:

- ASCII characters a-z, A-Z, period (.), and hyphen (-) are allowed. The first character of a name is always in the set a-z or A-Z.
- A name can have a maximum of 12 characters.
- Entity names are not case-sensitive, which means that `&body`, `&BODY`, and `&Body` are equivalent.

Besides the basic set of entities recognized in the text of an HTML skeleton story form, additional entities have been added to NRCS (version 1.3 and later) to allow for additional functionality.



*Do not forget to precede the entity with an ampersand and close it with a semicolon, such as `&F-TITLE`; or `&DEFINE:P;"BR"`*

The story entities are listed here:

Entity	Explanation
<code>&amp;BODY</code> ;	Interpolate the story BODY section in place of the entity. [false if story body section is empty] [true if story body section is not empty]
<code>&amp;AESET</code> ;	Interpolate the story AESET section in place of the entity. [false if story areset section is empty] [true if story areset section is not empty]

Entity	Explanation
&F-<name>;	<p>Interpolate the named field from the story in place of the entity.</p> <p>If the entity is one of F-AIR-DATE, F-CREATE-DATE, F-MODIFY-DATE, F-AUDIO-TIME, F-BACK-TIME, F-CUME-TIME, F-TOTAL-TIME, or F-TAPE-TIME, the field is output using the UNIX <code>strftime</code> function.</p> <p>Furthermore, a formatting string can follow these fields. The formatting string must immediately follow the ';' of the entity and be enclosed in single or double quote characters.</p> <p>The default format for the *-DATE fields is "%D %T". The default format for the *-TIME fields is "%M:%S".</p> <p>[true if the field is present in the story] [false if the field is not present in the story or is blank or empty]</p>
&IF:<entity-name>; Test <entity-name>	<p>If the entity yields a true condition, process template text up to a closing ELSE or ENDIF in the output stream. If the entity yields a false condition, skip over all template text up to the next ELSE or ENDIF (not output).</p> <p>Empty and blank fields are treated as a false IF condition. Only fields which are present and contain at least one non-blank character cause a true IF condition.</p>
&ELSE;	<p>Terminates the first part of an IF condition. If the IF condition was "true," it is now treated as "false" and all template text up to the closing ENDIF is skipped. If the IF condition was "false," it is now treated as "true" and all template text up to the closing ENDIF is processed normally.</p>
&ENDIF;	<p>Terminates an IF condition or an IF/ELSE combination.</p>

Entity	Explanation
<code>&amp;F-&lt;name&gt;;</code>	<p>Interpolate the named field from the story in place of the entity.</p> <p>If the entity is one of <code>F-AIR-DATE</code>, <code>F-CREATE-DATE</code>, <code>F-MODIFY-DATE</code>, <code>F-AUDIO-TIME</code>, <code>F-BACK-TIME</code>, <code>F-CUME-TIME</code>, <code>F-TOTAL-TIME</code>, or <code>F-TAPE-TIME</code>, the field is output using the UNIX <code>strftime</code> function.</p> <p>Furthermore, a formatting string can follow these fields. The formatting string must immediately follow the <code>';</code> of the entity and be enclosed in single or double quote characters.</p> <p>The default format for the <code>*-DATE</code> fields is <code>"%D %T"</code>. The default format for the <code>*-TIME</code> fields is <code>"%M:%S"</code>.</p> <p>[true if the field is present in the story] [false if the field is not present in the story or is blank or empty]</p>
<code>&amp;IF:&lt;entity-name&gt;;</code> <code>Test &lt;entity-name&gt;</code>	<p>If the entity yields a true condition, process template text up to a closing <code>ELSE</code> or <code>ENDIF</code> in the output stream. If the entity yields a false condition, skip over all template text up to the next <code>ELSE</code> or <code>ENDIF</code> (not output).</p> <p>Empty and blank fields are treated as a false IF condition. Only fields which are present and contain at least one non-blank character cause a true IF condition.</p>
<code>&amp;ELSE;</code>	<p>Terminates the first part of an IF condition. If the IF condition was "true," it is now treated as "false" and all template text up to the closing <code>ENDIF</code> is skipped. If the IF condition was "false," it is now treated as "true" and all template text up to the closing <code>ENDIF</code> is processed normally.</p>
<code>&amp;ENDIF;</code>	<p>Terminates an IF condition or an IF/ELSE combination.</p>

Entity	Explanation
&FIELDS;	Only used for test conditions [false if story fields section is empty or blank] [true if story fields section is not empty]

## NSML to HTML Conversion

The NSML specification is published on iNews' Web site at <http://www.avstarnews.com>

Avstar NRCS uses the News Story Markup Language (NSML) as its native file storage format. The txnet program automatically converts the NSML into HTML as stories are exported.

Various NSML tags can be redefined on export and translated to another code, if desired. For instance, it would be possible to translate the NSML paragraph tag into an HTML break tag (<BR>) instead of the default HTML paragraph tag (<P>).

By placing DEFINE statements at the top of the HTML skeleton story form before the <HTML> line, the default conversion of NSML tags into HTML tags can be modified. Format of a DEFINE statement is:

Single or double quotes can be used.

```
&DEFINE:<tag>;"<string>"
```

-OR-

```
&DEFINE;<tag>;'<string>'
```

The string includes all characters up to the matching delimiting quote; this allows inclusion of the other quotation mark within the string.

Standard definitions, which can be redefined, are listed in the following table:

**Table 15-1 NSML Tags and Output Strings**

Opening NSML Tag	Output String	Closing NSML Tag	Output String
P	"\r\n<p>"	/P	-----"</p>"
I	"<i>"	/I	-----"</i>"
U	"<u>"	/U	-----"</u>"
B	"<b>"	/B	-----"</b>"
PI	"<i>"	/PI	-----"</i>"
CC	"<b><i>"	/CC	-----"</i></b>"
A	"\r\n<!--a-->"	/A	-----"\n<!--/a-->"
AE	"\r\n<!--ae-->"	/AE	-----"\r\n<!--/ae-->"
AP	"\r\n<p>"	/AP	"</p>"
MC	"\r\n<!--mc-->"	/MC	"\r\n<!-- /mc -->"
IMG	"\r\n<!--img-->"		
PB	"\r\n<!--pb-->"		
TAB	"\r\n<!--tab-->"		
WP	"\r\n<!--wp-->"		

For instance, to redefine the open paragraph conversion, use:

```
&DEFINE:P;"output this".
```

To redefine the close paragraph conversion, use:

```
&DEFINE:/P;"found a close paragraph tag".
```



*The delimiting quotation mark must immediately follow the end of the entity, namely the semicolon.*

To include a carriage return or a line feed, use `\r` and `\n` in the `<string>`, as shown in Table 15-1 on page 15-11.

The display format for date or time fields can also be modified by a `DEFINE` statement in the HTML skeleton form:

- `&DEFINE:<type>;"<format>"`  
-OR-  
`&DEFINE;<type>;'<format>'`

Define default formats used to convert AIR-DATE, CREATE-DATE, and MODIFY-DATE data into date strings, and formats used to convert AUDIO-TIME, BACK-TIME, CUME-TIME, TAPE-TIME, and TOTAL-TIME data into time strings. The format includes all characters up to the matching delimiting quote (this allows inclusion of the other quotation mark within the format).

The “type” names used and default “formats” are:

```
DATE    "%D %T"
```

```
TIME    "%M:%S"
```

Format strings are defined by the UNIX/Irix `strftime` function.

You can still include a format string to override the default when referencing any of these fields individually, as in:

```
&f-create-date;"%C"
```



- `&SHOW:A;`

Causes output of tags for locating and referencing anchors and production cues (anchored elements). This is the default, which is the opposite functionality prior to introduction of this entity.

The string produced whenever an NSML `<a>` tag is encountered in the “template” is:

```
<A NAME="A<id>" HREF="#AE<id>">[<id>]</A>
```

For an NSML `<ae>` tag, the string produced is:

```
<A NAME="AE<id>" HREF="#A<id>">[<id>]</A>
```

The `<id>` is the production cue identifier, typically a small number.

This results in strings such as “[1]” appearing as HTML links in the text at the anchor point and also preceding the production cue. The two links point at each other.

- `&HIDE:A;`

Suppresses output of HTML anchor tags.

Empty fields and non-present fields in the HTML output are noted with comments similar to the following:

```
<!-- "var-1" field is empty -->
<!-- "app1-1" field not present -->
```

- `&HIDE:COMMENTS;`

If the `&HIDE:COMMENTS;` entity is processed, comments normally generated for the conditions “field not present” and “field is empty” will not be output to the HTML document.

This does not affect comments generated from error conditions.

- `&SHOW:COMMENTS;`

Generation of these HTML comments can be turned back on using `&SHOW:COMMENTS;`.

## Web Story Directives

Any story can contain Web Story directives. Directives are included as text. This allows system administrators to embed HTML codes within stories. Without these directives, characters and symbols such as the greater than sign (>) would normally be translated to `&gt;`; and not appear as HTML coding when exported to the Web server.

Directives	Explanation
<code>#HTMLSTART#</code>	All text in the story after this directive is processed as literal text. The characters '<', '>', '&', and "'" are output without being converted into HTML entities. Furthermore, all NSML tags are stripped from the input stream so they are <i>not</i> converted. However, to make viewing the HTML source a little easier, each paragraph closing tag is converted and output as a Carriage-Return / Line-Feed pair.
<code>#HTMLEND#</code>	This will terminate the processing of story text as literal text.

---

## Using Optional Format Strings

Optional format strings apply to date and time-related fields that you use for the field ID story entity references. You can add several time or date elements together (after the field ID) to create an optional format string, which changes the presentation of the time or date information field.



*The optional time format strings must be enclosed within quotes in the HTML skeleton story.*

## Time and Date Elements

The time and date elements to use in an optional format string to modify date and time field ID areas are:

**Table 15-2 Time and Date Elements**

<b>Time or Date Element</b>	<b>Description</b>
%%	Same as %
%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name
%c	Appropriate date and time representation
%C	Locale-specific century
%d	Day of month (01 - 31)
%D	Date as %m/%d/%y
%e	Day of month (1-31; single digits are preceded by a blank)
%h	Abbreviated month name
%H	Hour (00 - 23)
%I	Hour (01 - 12)
%j	Day number of year (001 - 366)
%m	Month number (1 - 12)
%M	Minute (00 - 59)
%n	Same as new-line

<b>Time or Date Element</b>	<b>Description</b>
%p	Equivalent of either AM or PM
%r	Time as %I:%M:%S [AM   PM]
%R	Time as %H:%M
%S	Seconds (00 - 61); allows for leap seconds
%t	Same as tab
%T	Time as %H:%M:%S
%U	Week number of year (00 - 53) (Sunday is the first day of week 1)
%w	Weekday number (0 - 6), Sunday is 0
%W	Week number of year (00 - 53) (Monday is the first day of week 1)
%x	Appropriate date representation
%X	Appropriate time representation
%y	Year within century (00 - 99)
%Y	Year as cyy (for instance, 1986)
%Z	Time-zone name or no characters, if time zone does not exist

The difference between %U and %W is which day is considered the first day of the week. Week number 01 is the first week in January, with a Sunday for %U or a Monday for %W. Week number 00 contains those days before the first Sunday or Monday in January for %U and %W, respectively.

## Time and Date Fields

Only certain fields—those relating to time and date—can recognize the optional format string, made up of the time and date elements. Specify these fields in field IDs, using the types shown here:

See “Avstar Form Fields” on page 8-13 for more information.

**Table 15-3 Time and Date Fields**

Field Type	Description	Default Format
MODIFY-DATE	Seconds since January 1, 1970 00:00:00 GMT	%b %e, %Y %T
CREATE-DATE	Seconds since January 1, 1970 00:00:00 GMT	%b %e, %Y %T
TOTAL-TIME	Total story time in seconds, the sum of audio and tape time	%M:%S
AUDIO-TIME	Audio read time of the story in sec- onds. Normally based on read rate and word count, but can be entered by the user	%M:%S
TAPE-TIME	Runtime, in seconds of a tape to be played with the story	%M:%S
BACK-TIME	Hard in-time of the story, in seconds	%M:%S
CUME-TIME	Hard out-time of the story, in seconds	%M:%S

If you use field names in the field IDs without creating an optional format string, default formats for the field types, as listed in the “Default Format” column in Table 15-3, are used.

### Time and Date Field Example

For instance, you decide to include a story creation date in the HTML story form. Instead of using defaults provided for the CREATE-DATE field (%b - abbreviated month name, %e - day of month, %Y - Year as

ccyy, %T - time as %H:%M:%S), you want the creation date information to appear differently.

1. Choose the following date and time elements:
  - %a – Abbreviated weekday name
  - %B – Full month name
  - %d – Day of month (01 - 31)
  - %Y – Year as ccyy
  - %r – Time as %I:%M:%S [AM | PM]
2. Enter elements in the optional format string, enclosing the string within quotes.

The `field_ID` story entity reference looks like this:

```
&f-create-date;"%a%B%d%Y%r"
```

The `field_ID` is embedded in HTML format tags. If the HTML story form is applied, the entry looks like this:

```
<b>CREATE-DATE:</b>&f-create-date;"%a%B%d%Y%r"
```

Data from the field would appear on the page similar to this:

```
CREATE-DATE: Mon July 10 2000 5:30:45 PM
```

## A Sample HTML Story Form

The following sample shows a news story in an Avstar HTML form used to define the appearance of the story, and the resulting HTML output.

### Characteristics of the Story

The story has the following characteristics:

- TITLE field with an associated value `TEASE-BREAK`
- MODIFY-DATE field with an associated value `862347754`
- PRESENTER field with an associated value `SB*`

- PAGE-NUMBER field with an associated value 75
- Two production cues (each formatted in its own paragraph):  
VO ENG  
Coming Up Next...
- Story text:  
(BELL)  
STILL AHEAD ON EYEWITNESS NEWS...  
THE MAXWELL CLUB HONORS SOME OF OUR AREA'S BEST  
HIGH SCHOOL FOOTBALL PLAYERS. YOU'LL MEET THEM...  
COMING UP.

## Contents of the Form

An HTML story form is applied to the story. The form's contents look like this:

```
<html>
<head>
<title>&f-title;</title>
<h1>
&f-title;
</h1>
</head>
<body>
<!--This is an example of an Avstar HTML form.-->
<table border="4" width="100%">
  <tr>
    <td colspan=1 align="left" width="10%">
      <b>PAG:</b>&f-PAGE-NUMBER;
    </td>
    <td colspan=1 align="left" width="40%">
      <b>MOD-DATE:</b>&f-modify-date;%D %T
    </td>
    <td colspan=1 align="left" width="50%">
      <b>PRESENTER:</b>&f-PRESENTER;
    </td>
  </tr>
  <tr>
<h2><th colspan=2 align="left" align="left">Story:
</th></h2>
  </tr>
  <tr>
    <td colspan=2 align="left" width="100%">
<p>&BODY;
```



```
        </td>
    </tr>
    <tr>
    <h2><th colspan=2 align="left" >Anchored Element
    List:</th></h2>
    </tr>
    <tr>
        <td colspan=2 align="left" width="100%">
            &AASET;
        </td>
    </tr>
</table>
</body>
</html>
```

## Resulting HTML Output

```
<html>
<head>
<title>TEASE/BREAK</title>
<h1>
TEASE/BREAK
</h1>
</head>
<body>
<!--This is an example of an Avstar HTML form.-->
<table border="4" width="100%">
  <tr>
    <td colspan=1 align="left" width="50%">
      <b>PAG:</b>75
    </td>
    <td colspan=1 align="left" width="40%">
      <b>MOD-DATE:</b>04/29/97 14:02:34
    </td>
    <td colspan=1 align="left" width="50%">
      <b>PRESENTER:</b>SB*
    </td>
  </tr>
  <tr>
<h2><th colspan=2 align="left" align="left">Story:
</th></h2>
  </tr>
  <tr>
    <td colspan=2 align="left" width="100%">
      (BELL)
      STILL AHEAD ON EYEWITNESS NEWS...
      THE MAXWELL CLUB HONORS SOME OF
```

---

OUR AREA'S BEST HIGH SCHOOL FOOTBALL  
PLAYERS. YOU'LL MEET THEM... COMING  
UP.

```
</td>
</tr>
<tr>
<h2><th colspan=2 align="left" >Anchored Element
List:</th></h2>
</tr>
<tr>
<td colspan=2 align="left" width="100%">
<p><I>VO ENG</p>
<p>Coming Up Next...
</td>
</tr>
</table>
</body>
</html>
```



# CHAPTER 16

## *Web Access*

Web Access allows direct, read-only browsing of the Avstar NRCS database through a Web browser.

This chapter includes:

- Overview
- Starting the Web Server
  - Web Access Login
- Web Access Story Templates
- Web Access Directory and Queue Templates
  - Template Entities
- Web Access Configuration

## Overview

The Web Access software is normally installed with your Avstar Newsroom Computer System. Avstar NRCS:

- Accesses the database
- Formats the story the user selects into Hypertext Markup Language (HTML), using the default HTML story form
- Presents the story to the user's Web browser

Format of information sent to the user's Web browser is configurable by the system administrator. How the directory listing, queue contents, and stories are converted to HTML for presentation are governed by HTML templates or Web skeleton stories inside the database.

## Starting the Web Server

The Apache Web server is loaded on Avstar Servers with normal installation of Avstar NRCS. The commands to start the Web server and Web Access server programs are:

```
/usr/local/etc/httpd/httpd  
/etc/webaccserver
```

These commands are usually placed in the `/site/exc/connect.sh` command script, which is executed whenever servers are connected (or reconnected). You can check to see if the Apache Web server is currently running on your system by pointing your browser to:

```
http://<servername>
```

where `servername` is the Avstar Server name you normally log in to from the Avstar Workstation. If the Web server is running, you should see a screen with the Apache logo that says "it worked." To customize it for your site, this page is located at:

```
/usr/local/etc/httpd/htdocs/index.html
```

## Web Access Login

You must have at least one Web session configured on your system to log in over the Web. The format of a Web session resource in the `/site/config` file is:

```
websession          <session number>
```

such as:

```
websession          401
```

To log in to the Avstar database with a Web browser, point the browser to the following URL:

```
http://<servername>/cgi-bin/NewsWeb
```

This will open a page containing a drop-down list of Avstar Servers accessible via the Web. When you select one of those servers and click submit, you will be taken to a login screen. If you know the servers configured for Web Access, you may include their names in the URL to go directly to the login dialog page:

```
http://<servername>/cgi-bin/NewsWeb.avstar-a
```

At the login dialog screen, enter your Avstar user name and password and click login.

When logging in over the Web for read-only database access, group security on the system is respected. Users who do not have access to queues and directories protected by a read-group will not see those areas when they log in over the Web.



*The message of the day screen is not displayed when logging in via a Web browser.*

## Web Access Story Templates

When users view a story in their Web browser, the system presents the story in HTML format. The appearance of this story is configurable by the system administrator via a Web Access form or HTML skeleton

story form, similar to the HTML skeleton stories used for Web publishing.

When accessing over the Web, the story form assigned to a story is noted and a corresponding Web Access form is searched for under queues specified by the `Q_WEBACC_FORMS` entry in the system's queues dictionary. The default location for the Web Access form (HTML skeleton story form) is `SYSTEM.WEBFORMS`.

The appropriate Web Access form is sought from `SYSTEM.WEBFORMS.<first letter of story form>.<story form>`.

For instance, if the form assigned to the story was called `script`, then the Web Access form—that is, the top HTML skeleton story—in the queue, `SYSTEM.WEBFORMS.S.SCRIPT` would be used to display the story. Stories assigned the `rundown` story form would be displayed with the Web Access form in `SYSTEM.WEBFORMS.R.RUNDOWN`. Stories assigned the `memo` story form would look for the top HTML skeleton story in the queue, `SYSTEM.WEBFORMS.M.MEMO`, for their display on the Web browser.

If a corresponding Web Access form was not created, the system uses the HTML skeleton story form in the default Web Access queue as specified by the `W_WEBACC_FORM` entry in the queues dictionary. The default value is `ACCESS-FORM`. So, a story assigned a form that does not have a corresponding `SYSTEM.WEBFORMS` queue will be displayed with the system-wide default HTML skeleton story form in `SYSTEM.WEBFORMS.A.ACCESS-FORM`.



*If a system-wide default Web Access form has not been created, the story will be displayed with a hard-coded default.*

The following Web Access form locations will be checked, in order, for an HTML skeleton story form to display stories:

1. `SYSTEM.WEBFORMS.<first letter of story form>.<story form>`
2. `SYSTEM.WEBFORMS.A.ACCESS-FORM`
3. Hard-coded internal default



You may create HTML skeleton stories to define the appearance of stories viewed via a Web browser. The form is stored as an Avstar database story whose text is in HTML with optional embedded references to story entities. Avstar NRCS contains an HTML skeleton story form to use as the default form, but you can customize it or create new forms, using story entities. A full listing of all entities available to Web Access forms used for database browsing appears in “Template Entities” on page 16-8.



*Chapter 15 reviews story entity reference options in greater detail. The same HTML skeleton story forms set up on your system for exporting stories in HTML format can also be used when browsing the database over the Web. See “Adding Story Entity References” on page 15-5 for more information.*

## Default Story Template

The following is the default story template, provided with the Avstar NRCS software.

```
<html>
<head>
<title>&F-TITLE;</title>
</head>
<body>
<table border=4 width='100%'>
  <tr>
    <td colspan=1 width='50%'><b>CREATED BY:</b>&F-CREATE-BY;
    </td>
    <td colspan=1 width='50%'><b>CREATED:</b>&F-CREATE-DATE;
    </td>
  </tr>
  <tr>
    <td colspan=1 width='50%'><b>PAGE:</b>&F-PAGE-NUMBER;
    </td>
    <td colspan=1 width='50%'><b>MODIFIED:</b>&F-MODIFY-DATE;
    </td>
  </tr>
  <tr><h2>
    <th align=left colspan=2>Story:
    </th></h2>
  </tr>
  <tr>
    <td colspan=2 width='100%'>&BODY;
    </td>
  </tr>
  <tr><h2>
    <th align=left colspan=2>Anchored Element List:
    </th></h2>
  </tr>
  <tr>
    <td colspan=2 width='100%'>&AESET
    </td>
  </tr>
</table>
</body>
</html>
```

## Web Access Directory and Queue Templates

Avstar NRCS version 1.3 allows the system administrator to customize the display of directory listings and listings of a queue's contents via system-wide directory and queue HTML skeleton stories (also known as templates) rather than predefined formats. However, it is not necessary to customize directory and queue templates. Defaults can be used, if desired. See "Default Directory Template" on page 16-18 and "Default Queue Template" on page 16-19 for more information.

The template used for directory pages is located in the queue defined by the `Q_WEBACC_FORMS` and `W_WEBACC_FORM` definitions (`SYSTEM.WEBFORMS.A.ACCESS-FORM`). To distinguish this directory template from the story template, the directory template is identified by having the word `directory` as the first word in the index field (also known as the sortfield). Only a single template is required for directory page construction.

The template used for queue pages is located in the queue defined by the `Q_WEBACC_FORMS` definition and the queue form name assigned to the queue being displayed. This is similar to how queue forms are handled. To identify the template as a queue template, the word "queue" must appear as the first word in the index field (also known as the sortfield). This allows both the queue template and story template with the same name to reside in the same queue.

If a queue template is not found using the queue form name of the queue, the system looks in the queue defined by the `Q_WEBACC_FORMS` and `W_WEBACC_FORM` definitions, similar to story templates. The queue template is identified with the word "queue" as the first word in the index field.

Put the system-wide default Web Access form (HTML skeleton story form) at the top of the `SYSTEM.WEBFORMS.A.ACCESS-FORM` queue. Place stories titled `directory` and `queue` beneath the form used by default to display story contents.

## Template Entities

The following table lists all of the available entities that can be placed in either a story, directory, or queue template. The Q- and D- entities are only valid for Web Access. They are not available for Web Publishing. If an entity is not valid for a certain type of template, it is noted in the entity description.

Each entity now has IF condition definitions included following the text handling description. If an entity is not valid for a certain type of template, it will always result in a false condition when tested. Empty and all blank fields produce a false IF condition. Only fields present in the story and containing at least one non-blank character cause a true IF condition.



*Do not forget to precede the entity with an ampersand and close it with a semicolon, such as `&F-TITLE;` or `&DEFINE:P;"BR"`*

Entity	Explanation
<code>&amp;BODY;</code>	Interpolate story BODY section in place of entity. [false if story body section is empty] [true if story body section is not empty]
<code>&amp;AESET;</code>	Interpolate story AESET section in place of entity. [false if story areset section is empty] [true if story areset section is not empty]

Entity	Explanation
&F-<name>;	<p>Interpolate the named field from the story in place of the entity.</p> <p>If the entity is one of F-AIR-DATE, F-CREATE-DATE, F-MODIFY-DATE, F-AUDIO-TIME, F-BACK-TIME, F-CUME-TIME, F-TOTAL-TIME, or F-TAPE-TIME, the field is output using the UNIX <code>strftime</code> function.</p> <p>Furthermore, a formatting string can follow these fields. It must immediately follow the ‘;’ of the entity and be enclosed in single or double quotation marks.</p> <p>Default format for the *-DATE fields is “%D %T”  Default format for the *-TIME fields is “%M:%S”</p> <p>[true if the field is present in the story]  [false if the field is not present in the story or is blank or empty]</p>
&IF:<entity-name>; Test <entity-name>	<p>If the entity yields a true condition, process template text up to a closing ELSE or ENDIF in the output stream. If the entity yields a false condition, skip over all template text up to the next ELSE or ENDIF (not output).</p> <p>Empty and blank fields are treated as a false IF condition. Only fields which are present and contain at least one non-blank character cause a true IF condition.</p>
&ELSE;	<p>Terminates the first part of an IF condition. If the IF condition was “true,” it is now treated as “false” and all template text up to the closing ENDIF is skipped. If the IF condition was “false,” it is now treated as “true” and all template text up to the closing ENDIF is processed normally.</p>
&ENDIF;	<p>Terminates an IF condition or an IF/ELSE combination</p>


Entity	Explanation
&FIELDS;	<p>Only used for test conditions</p> <p>[false if story fields section is empty or blank]</p> <p>[true if story fields section is not empty]</p>
&REPEAT;	<p>Bracket a section of the template text, up to an ENDREPEAT entity. The bracketed text will be processed for each object to be displayed on the page. This is only valid for directory and queue templates. If there are no objects to be displayed, the directory or queue is empty, and all of the bracketed text is skipped over.</p>
&ENDREPEAT;	<p>Terminates a REPEAT section. If there is another subdirectory or queue entry to be displayed on the page, template processing will continue at the preceding REPEAT entity. If there are no more objects to be displayed, template processing continues at the point following this entity.</p>
&D-CHILD;	<p>Interpolate current subdirectory name in place of the entity. Not valid for queue or story templates.</p> <p>[always true]</p>
&D-COUNT;	<p>Interpolate number of sub-directories contained in current directory. This is intended primarily for test conditions, to determine if a directory is empty. Not valid for queue or story templates.</p> <p>[false for empty directory]</p> <p>[true for directory with at least one entry]</p>
&D-CURRENT;	<p>Interpolate a link reference, such as <code>HREF</code>, to current sub-directory entry in place of the entity. Not valid for queue or story templates.</p> <p>[false for an empty directory]</p> <p>[true for a directory with at least one subdirectory]</p>


Entity	Explanation
&D-DIRECTORY;	<p>Nothing, only used for test condition. Not valid for queue or story templates. This is the opposite of D-QUEUE.</p> <p>[true if the current subdirectory is a directory] [false if the current subdirectory is a queue]</p>
&D-LOGOUT;	<p>Interpolate a link reference, such as <code>HREF</code>, which includes a Web Access server logout request in place of the entry.</p> <p>[always true]</p>
&D-NAME;	<p>Interpolate the current directory name in place of the entity.</p> <p>[always true]</p>
&D-PARENT;	<p>Interpolate a link reference, such as <code>HREF</code>, to the parent directory of the current directory in place of the entity.</p> <p>[false when current directory is the root directory] [true when current directory is not the root directory]</p>
&D-QUEUE;	<p>Nothing, only used for test condition. Not valid for queue or story templates. This is the opposite of D-DIRECTORY.</p> <p>[true if the current subdirectory is a queue] [false if the current subdirectory is a directory]</p>
&D-SERVER;	<p>Interpolate the server name (computer name) in place of the entity.</p> <p>[always true]</p>

Entity	Explanation
&Q-COUNT;	<p>Interpolate number of queue entries contained on current page. This is intended primarily for test conditions, to determine if a queue is empty. Not valid for directory templates.</p> <p>[false for empty queue] [true for queue with at least one entry] [always true for story template]</p>
&Q-CURRENT;	<p>Interpolate a link reference, such as <code>HREF</code>, to current queue entry in place of the entity. This includes the complete URL-string to identify the current story position within the queue. Not valid for directory templates.</p> <p>[false for empty queue] [true for queue with at least one entry] [always true for story template]</p>
&Q-HREF;	<p>alias for Q-CURRENT (preferred name)</p>
&Q-LOGOUT;	<p>Interpolate a link reference, such as <code>HREF</code>, which includes a Web Access server logout request in place of the entry.</p> <p>[always true]</p>

---



Entity	Explanation
&Q-MODIFY-DATE;	<p>Interpolate modified time of current queue entry in place of the entity. Not valid for directory templates. The modified time is formatted using the <code>strftime</code> function with a default format definition of “%D %T”. This format can be redefined via the <code>DEFINE:DATE</code> entity described below. Just as with *-DATE fields, you can supply a format following this entity.</p>
	<p><i>This is the queue entry modified time and NOT the story modified time.</i></p>
&Q-NAME;	<p>[always true]</p>
&Q-NAME;	<p>Interpolate current queue name in place of the entity. [always true]</p>
&Q-NEXT;	<p>Interpolate a link reference, such as <code>HREF</code>, to the next queue entry within the queue. This includes the complete URL-string to identify the next queue entry position within the queue. Not valid for directory templates.</p> <p>[false in queue template when current page contains last queue entry] [true in queue template when current page does not contain last queue entry] [false in story template when current story is the last story in the queue] [true in story template when current story is not the last story in the queue]</p>
&Q-PARENT;	<p>Interpolate a link reference, such as <code>HREF</code>, to parent directory of current queue in place of the entity. [always true]</p>

Entity	Explanation
&Q-PREV;	<p>Interpolate a link reference, such as <code>HREF</code>, to the previous queue entry within the queue. This includes the complete URL-string to identify the previous queue entry position within the queue. Not valid for directory templates.</p> <p>[false in queue template when current page contains first queue entry]</p> <p>[true in queue template when current page does not contain first queue entry]</p> <p>[false in story template when current story is the first story in the queue]</p> <p>[true in story template when current story is not the first story in the queue]</p>
&Q-QUICK-INDEX;	<p>Interpolate quick index contents of current queue entry. Not valid for directory templates.</p> <p> <i>Twenty (20) characters are always interpolated. This is different than story field contents where only the text present is interpolated.</i></p> <p>[always true]</p>
&Q-SERVER;	<p>Interpolate the server name (computer name) in place of the entity.</p> <p>[always true]</p>
&Q-STORY-TIME;	<p>Interpolate story read time of current queue entry in place of the entity. Not valid for directory templates. The modified time is formatted using the <code>strftime</code> function with a default format definition of “%M:%S”. This format can be redefined via the <code>DEFINE:TIME</code> entity described below. Just as with <code>*-TIME</code> fields, you can supply a format following this entity.</p> <p>[always true]</p>

Various News Story Markup Language (NSML) tags can be redefined and translated to another code, if desired, by placing DEFINE statements at the top of the HTML skeleton story form before the <HTML> line. The format of a DEFINE statement is:

```
DEFINE:<tag>;"<string>"
```

-OR-

```
DEFINE;<tag>;'<string>'
```

Single or double quotes can be used. The string includes all characters up to the matching delimiting quote (this allows inclusion of the other quotation mark within the string). Standard definitions, which can be redefined, are listed in the following table.

<b>Opening NSML Tag</b>	<b>Output String</b>	<b>Closing NSML Tag</b>	<b>Output String</b>
P	"\r\n<p>"	/P	----"</p>"
I	"<i>"	/I	----"</i>"
U	"<u>"	/U	----"</u>"
B	"<b>"	/B	----"</b>"
PI	"<i>"	/PI	----"</i>"
CC	"<b><i>"	/CC	----"</i></b>"
A	"\r\n<!-- a -->"	/A	----"\n<!-- /a -->"
AE	"\r\n<!-- ae -->"	/AE	----"\r\n<!-- /ae -->"
AP	"\r\n<p>"	/AP	"</p>"
MC	"\r\n<!-- mc -->"	/MC	"\r\n<!-- /mc -->"
IMG	"\r\n<!-- img -->"		
PB	"\r\n<!-- pb -->"		
TAB	"\r\n<!-- tab -->"		
WP	"\r\n<!-- wp -->"		

For instance, to redefine the open paragraph conversion, use:

```
&DEFINE:P;"output this"
```

To redefine the close paragraph conversion, use:

```
&DEFINE:/P;"found a close paragraph tag"
```



*The delimiting quote mark must immediately follow the end of the entity, namely the semicolon. Also, to include a carriage return or a line feed, use `\r` and `\n` in the `<string>`, as shown in the table.*

The display format for date or time fields can also be modified by a `DEFINE` statement in the HTML skeleton story form:

```
DEFINE : <type> ; " <format> "
```

Define default formats used to convert AIR-DATE, CREATE-DATE, and MODIFY-DATE data into date strings, and formats used to convert AUDIO-TIME, BACK-TIME, CUME-TIME, TAPE-TIME, and TOTAL-TIME data into time strings. The format includes all characters up to the matching delimiting quote (this allows inclusion of the other quotation mark within the format).

The “type” names used and default “formats” are:

```
DATE      "%D %T"
TIME      "%M:%S"
```

If you want to alter the default display format of date and time values, see “Using Optional Format Strings” on page 15-14 for more information.

You can still include a format string to override the default when referencing any of these fields individually, as in:

```
&f-create-date ; "%C"
```

```
SHOW : A
```

Causes output of tags for locating and referencing production cues (anchored elements). This is the default, which is the opposite functionality prior to the introduction of this entity.

The string produced whenever an NSML `<a>` tag is encountered in the “template” is:

```
<A NAME="A<id>" HREF="#AE<id>">[<id>]</A>
```

For an NSML `<ae>` tag, the string produced is:

```
<A NAME="AE<id>" HREF="#A<id>">[<id>]</A>
```

The `<id>` is the anchored element identifier, which is typically a small number.

This results in strings such as “[1]” appearing as HTML links in the text at the anchor point and also preceding the anchored element. The two links point at each other.

`HIDE:A` Suppresses output of HTML anchor tags.

## Default Directory Template

The following is the default directory template included with the Avstar NRCS software:

```

<html>
<head>
  &IF:D-PARENT;
  <title>&D-SERVER;</title>
&ELSE;
  <title>&D-SERVER;:&D-NAME;</title>
&ENDIF;
</head>
<body><a name=top></a>
<h3>
  &IF:D-PARENT;
  <img hspace=5 src='/icons/
folder.open.gif'>[&D-SERVER; ]&D-NAME;
&ELSE;
  <img hspace=5 src='/icons/comp.gray.gif'>[&D-SERVER;]
&ENDIF;
</h3>
  <img src='/icons/blank.gif' hspace=5><a
href=&D-LOGOUT;>[Logout]</a>
&IF:D-PARENT;
&nbsp;&nbsp;&nbsp;
<a href=&D-PARENT;>
  <img src='/icons/back.gif' hspace=5 border=0>Return to Parent
Directory
</a>
  &ENDIF;
  <hr>
&IF:D-COUNT;
&REPEAT;
  <a href=&D-CURRENT;>
    &IF:D-QUEUE;
    <img src='/icons/text.gif' hspace=5 border=0>
    &ELSE;
    <img src='/icons/folder.gif' hspace=5 border=0>
    &ENDIF;
    &D-CHILD;
  </a>
  <br>
&ENDREPEAT;
&ELSE;
  <img src='/icons/blank.gif' hspace=5>[Empty]<br>
&ENDIF;
</body>
</html>

```







---

printed on the console. This includes failures due to an unknown user name, an invalid password, or a *blacklisted* user.

`announce logins=<value>`

If the value is non-zero, all logins will be printed on the console.

`diagnostics=<value>`

This is primarily for debugging purposes. The value is a bitmask, which controls printing of diagnostics. For instance, if the value is odd, such as the 1 bit is on, a diagnostic is printed every time a session is requested but all session licenses are in use.

`hard expire = <seconds>`

Number of seconds of inactivity which will force a reauthentication by the user and release the Web session for another user to log in to.

`putenv=<environment string>`

Allows environment settings to be put into the initial environment. The environment string should be of the form `name=value`. This will be primarily used for debugging purposes.

`queue entries per page = <number>`

Number of stories to put on a queue page. This can be a number from 1 to 50.

`soft expire = <seconds>`

Number of seconds of inactivity before the session license can be reused for another session. This will only affect a user if the server gets a session request and does not have any free session licenses. The server attempts to obtain the oldest session license—that is, the one that has been inactive the longest. If that session has been inactive for more than the soft expire interval, it will be reused.

(Anytime a session is inactive for more than the hard expire interval, the user will be required to reauthenticate the session.)

The following definitions must be present in the file `/site/web/strings`, accessible by the Web Access server on the computer on which the Web Access server program is running. The format of this file is the same as all dictionary files. This file is read once when the Web Access server starts. Definitions can be localized, as necessary.

session error	/Avstar news session error
session login	/Avstar news session login
session terminated	/Your Avstar news session has been successfully terminated!
offline	/Avstar news is offline.
try again later	/Please try again later.
cannot set subdirectory	/Cannot set subdirectory.
cannot open dir file	/Cannot open directory file.
cannot open queue	/Cannot open queue.
cannot access	/Cannot access
invalid certificate	/Invalid session certificate
no license	/No Web Access licenses available.
name	/Name:
password	/Password:
login	/Login
clear login	/Clear Form
enter login	/Please enter your name and password

---

reauthenticate	/Avstar news Web session reauthentication
session expired	/Your Avstar news session authentication has expired
reenter login	/Please re-enter your user name and password.
not authorized	/You are not authorized to access the Avstar database
return to parent	/Return to parent directory
bad login	/The user name and/or password you entered are not valid.
no login see admin	/If you do not have a login, see your system administrator.
story unavailable	/You cannot access this story.
;	
;	Used by "NewsWeb" script
;	
no servers	/There are no Avstar Servers configured.
server session	/Avstar Server selection
server error	/Avstar Server selection Error
server admin	/See your system administrator
server submit	/submit
server select	/Please select an Avstar Server and click on "submit."



# SECTION III

## *System Operations and Troubleshooting*

This section contains the following chapters:

- Chapter 17, *Connect Services*
- Chapter 18, *Database Security*
- Chapter 19, *Database Management*
- Chapter 20, *Backing Up Your System*
- Chapter 21, *Disconnects*
- Chapter 22, *Troubleshooting*

---

# CHAPTER 17

## *Connect Services*

Connect services are utilities you can set up to enable Avstar users to connect to other computers or information services over the network, such as a remote console for your Avstar Newsroom Computer System (NRCS).

This chapter contains the following sections:

- Network Services
- Dialogs for Connect Services
  - Building a Dialog
- Adding System Services
  - Setting up the Service
- Console Connect Sessions
- Serial Connect Services

## Network Services

It is useful to set up a service to allow you to connect to your Avstar Servers. For instance, setting up network remote connect services allows you to perform many administrative tasks from your Avstar Workstation rather than at the Avstar console.

## Dialogs for Connect Services

A *dialog* is a script that tells the service what prompts to expect from the device it connects to, and provides the service with appropriate responses for each prompt. Using any service involves some routine activities, such as logging in, that you do each time you use that service. You can create a dialog for any service to handle these routine activities.

## Building a Dialog

This section explains how to design and create a simple dialog that logs a user in to an information service. Once logged in, the dialog yields control to the user until he/she is ready to close the connection. Then the dialog logs the user out of the service and closes the connection.

When you design a dialog, it is helpful to turn on the `capture connect` command and perform the procedure you want to incorporate into the dialog. This way, steps in that procedure are captured to a story that you can refer to while designing the dialog; each line in the dialog is built as a line of the story.

This process includes the most commonly used dialog commands.



To build a dialog, do the following:

1. Create a dialog queue.

Each dialog must be in a separate queue in the `SYSTEM.DIALOG` directory. The first step in creating a new dialog is to go to `SYSTEM.DIALOG` and create a queue to hold it. The name you give this queue is also the name of the dialog, so choose a queue name that describes what the dialog does, such as `Console`, for a dialog used during connect sessions to the Avstar console. See “Adding a Directory or Queue” on page 5-3 for more information.

2. Open a new story in the queue and build the dialog.
3. Type the `message` dialog command using the following format:

```
message <"string">
```

The `message` dialog command displays on the user's screen whatever text you put in the `<"string">` parameter.

Type a message that will be displayed on the user's workstation, indicating the connection is being established.

4. Type:

```
wait CONNECT
```

When you use a modem to make the connection, you want the dialog to wait until the modem sends the word `CONNECT` to the workstation, signaling it has made the connection.

If necessary, use these additional commands to accommodate the login procedure:

```
delay nn
```

Use this command to pause the dialog while the service continues connect and login processes.

(`nn` is the number of seconds you want the dialog to pause before continuing to the next command)

`type <"string">` Use this to send a message to the device the workstation is being connected to.  
`<"string">` includes the message

5. To send your login data, such as an account number, type:

`type <"string">`

6. Type `pass` to yield control to the user after login is complete.

This command instructs the dialog to pass whatever the user enters to the device to which the service has connected.

Use `pass x` to include a character the user can enter to cue the dialog to resume. The `x` should be a character or symbol, such as the at symbol (@), he or she will not normally use.

Use `pass` alone to instruct the system to accept what the user enters until he/she closes the connection.

7. Type commands to log out of the service.

When the user exits a connection using the `quit connect` command, the dialog resumes and performs the logout process and closes the connection.

8. Attach the dialog to the service by placing the dialog's name in the service's `dialog` parameter in the service table, located in the `SYSTEM.SERVICE` queue.
9. Type the `configure -s` command to incorporate this change into the service.

## Dialog Commands

Available dialog commands are flexible enough to script entire connect sessions. For instance, a dialog could be constructed to automatically log in to the remote console and "unbusy" a rundown if desired.

The available dialog commands are reviewed in Appendix A.

## Dialog Examples

Here are a few examples of dialogs for remote console connect services:

SLUG	CREATED
console	5/28/91 16:24:59

```
diag off
capture system.capture
;delay 10
pass|
```

In the last example, `<cr>` and `<lf>` are used to indicate a “cursor return” and “line feed.”

Slug	Tal	Created by	Modified
console		news	1/13/98 15:32:49

```
escape <26>
message Connecting to the console...<cr><lf>
message Remember: Do not restart devices from a connect session!<cr><lf>
message Logout by CTRL-D, not by closing window!<cr><lf>
capture system.capture
pass
```

## Adding System Services

A service consists of two parts:

- A network resource, set up as a device in the configuration file (`/site/config`).
- The service, set up in the service table in the database file (`SYSTEM.SERVICE`); it uses the resource to make a connection.

To add a new service to your system, you must design and build the service and select a network resource for the service to use. If an appropriate resource does not exist, create one.

## Setting up the Service

To set up the service and its resource, do the following:

1. See if the service and its resource exists.

Try connecting to a service you want. If it works, then the necessary programs are installed. If you cannot find the service, or one does not work, call iNews Customer Support for help.

2. Choose a device number for the resource and a name for the service. In the following examples, the service name is `console`.

Check your configuration file (`/site/config`) to determine the appropriate device number for your new resource. For a network resource, choose a number in the range you have reserved for services, servers, and similar devices. Ensure the number you choose is out of the range used for normal devices. See “PCU Device Numbering” on page 11-34 for more information.

3. Add the resource to the configuration file on each server in your system.

Add the device number of the network resource to the `reslist` line in the server’s host definition. To keep the network resource available if you bring down one of the system servers, add this line to any appropriate alternate host definitions. See “The Hosts File (`/etc/hosts`)” on page 11-17 for more information.

- a. Select *all* servers. See “Selecting Servers” on page 2-5 for more information.

- b. Use `ed` (UNIX line editor) to open the configuration file. The display will look similar to:

```
AVSTAR-A: ed /site/config
editing /site/config
1356
```

The general format for a network resource configuration line is:

```
resource <device #> <resource name> - <device name>
```

See “The Site Configuration File (`/site/config`)” on page 11-7 for more information.

The following steps use `ed` to modify system files. If you are unfamiliar with this UNIX line editor, see “Using the UNIX Editor” on page 10-1.

Parameter	Description
device #	Identifies resource's device number.
resource	Name you want to give to the resource. Network resources can share the same name; services choose the first available resource of the correct kind. In our example, we call the resource net.
device name	Resource's device name. If you do not want to give it a name, put a hyphen in this position.

For instance, our new network resource configuration line would look like this:

```
resource 220 console - - ; net connect
```

#### 4. Add the service to the service table in the database.

Each service installed in your system is defined on a separate line in your system's service table, which is the first story in `SYSTEM.SERVICE`. Each service defined in the service table consists of a few parameters that determine how the service behaves. It does not matter where in the service table you add the new line.

A service line has six parameters in this general format:

```
<service> <host> <dialog> <resource> <group> <command>
```

**Table 17-1 Service Line Format**

Parameter	Definition
service	Name you want the service to have.
host	Name of the server where you want the service to look for its resource. List the target server's name in your system's <code>etc/hosts</code> file.  Normally, put a hyphen in this position of the service line to force the service to search each server until it finds one that has a resource it can use.

Parameter	Definition
<code>dialog</code>	If you want the service to use a dialog, put the dialog name in this position of the service line. Otherwise, fill this position with a hyphen.
<code>resource</code>	Name of the resource that you want the service to use.
<code>group</code>	You can restrict who can use the service by specifying a user group. If you do not want to restrict access, place a hyphen in this position.
<code>command</code>	Specifies the command you want the service to use to handle communication. To create a service that establishes a network connection, type the <code>telnet</code> command here. Ensure to use its full pathname and include the system name to which you are connecting.

You will need one such service line for each server on your system. The pathname for your service will vary, depending on which platform supports your system.

#### SCO Systems

For SCO systems, use the `rlogin` command in the following way:

```
/usr/bin/rlogin <servername>
```

The following sample entry in `SYSTEM.SERVICE` lets you establish a network connection to your console from an Avstar Workstation session.

```
cona AVSTAR-A console net - /usr/bin/rlogin AVSTAR-A -l so
```

#### SGI Irix Systems

For SGI Irix systems, use the `login` command in the following way:

```
/bin/login <username>
```

The following sample entry in `SYSTEM.SERVICE` lets you establish a network connection to your console from an Avstar Workstation session.

```
cona - console neta - /bin/login so
conb - console netb - /bin/login so
```

The resource assigned to the service must be configured on the computer you want to log in to.

Here's an example of a service table in `SYSTEM.SERVICE`:

[AVSTAR]SYSTEM.SERVICE.service table					
SLUG	CREATED	MODIFIED	BY	TIME	
service table		01/26/00 10:31:53	avstar	0:20	
;					
;service host dialog resource group command					
;					
a	-	console	console	sysops	/usr/bin/rlogin a -l so
b	-	console	console	sysops	/usr/bin/rlogin b -l so
;					
; connect to console format, by platform:					
;					
SCO	-	console	console	sysops	/usr/bin/rlogin a -l so
SGI	<host>	console	console	sysops	/bin/login so
RISC	<host>	console	console	sysops	/bin/login so

5. (Optional) Create a dialog for the service.

If you want your service to follow certain instructions every time it is invoked, create a dialog for the service that contains those instructions. See “Building a Dialog” on page 17-2 for more information.

6. (Optional) Test your configuration changes on Avstar NRCS. See “Testing the Site Configuration File After Changing” on page 11-9 for more information.

7. Reconfigure the system.

This causes your system to note changes and incorporate them into appropriate programs. Do the following:

- a. Select the master computer (typically server A). See “Selecting Servers” on page 2-5 for more information.
- b. Become a superuser. See “Becoming a Console Superuser” on page 3-2 for more information.

c. Type:

```
AVSTAR-A: offline
```

d. If you added a new resource or modified an existing one in the process of creating a new service, reconfigure your system by typing the following:

```
AVSTAR-A# configure
```

If you modified an existing service or added a service that uses an existing network resource, you did not make any changes to your system's configuration file. So, you need to have the system note only changes made to the service table by typing the following:

```
AVSTAR-A: configure -s
```

8. When the prompt returns, type:

```
AVSTAR-A# online
```

9. Exit superuser mode by holding the Control key (CTRL) down and typing the letter D. A message similar to the following will appear:

```
AVSTAR-A# CTRL-D
```

```
A Tue Oct 5 00:18:58 2000 msg System being configured
```

When you see the system prompt, the network service you created and its resource you added (if any) is ready for use.

10. (Optional) Back up site files.

If you have made significant changes, back up your site files with the `sitedump` console command.

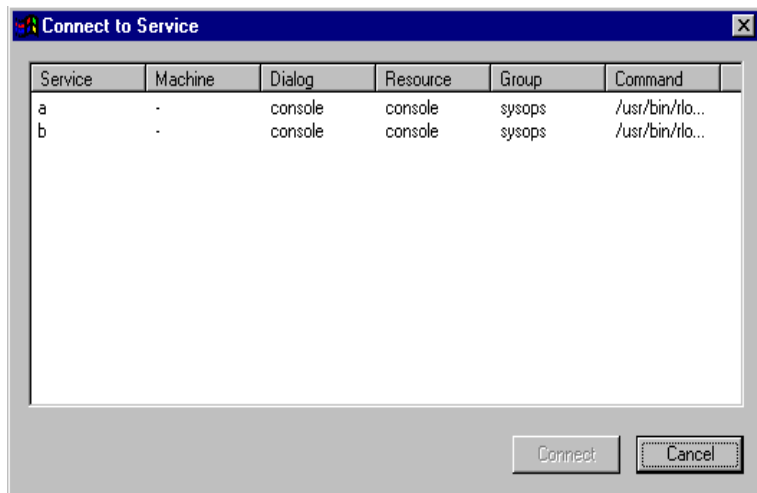


## Console Connect Sessions

To connect to the Avstar console from an Avstar Workstation—that is, once the network remote connect service and its resources are set up—do the following:

1. Log in to any Avstar Workstation, with a user account allowed to use the Connect to Service feature.
2. Click the Communicate drop-down menu.
3. Select Connect to Service. The Connect to Service dialog box will appear, offering you a list of services.

As shown in this example, two options are provided to connect to the Avstar console—each option corresponding to a different Avstar Server.



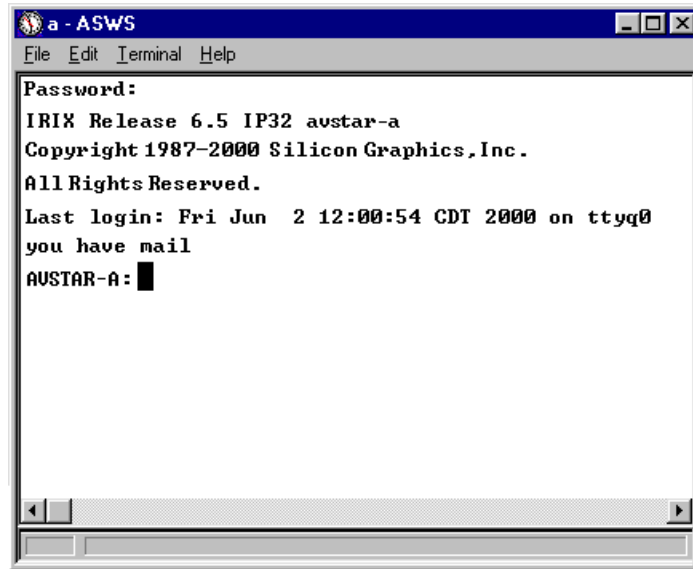
4. Double-click on the service you want.

A dialog box will appear with a dialog (as defined in the `SYSTEM.DIALOG`) that requests a password.

5. Type in the password and press Enter.

6. Once connected, the dialog box will display the console prompt, similar to what appears on the Avstar console when you are logged in as a system operator.

Here is an example of a remote console connection:



7. You can now perform various administrative tasks through the console connect session, as opposed to doing them while physically located at the Avstar console.



**Do not restart devices through a console connect session.**



*To stop a console connect session, press CTRL-D. Do not close the dialog box by selecting the Close option from the File drop-down menu or by clicking on the X button in the upper-right corner.*

## Serial Connect Services

Although less commonly used since the advent of the Internet, serial connect services can also be configured to connect and log in to other computers, either via modem dialup or direct serial line.

Contact iNews Customer Support if you need to configure a serial connection for your users.



# CHAPTER 18

## *Database Security*

This chapter describes how to use various features in Avstar NRCS to establish and maintain database security.

This chapter includes the following sections:

- Establishing Security Procedures
- User Passwords
- Tracking User Activity
- Using Group Security to Control System Access

## Establishing Security Procedures

Use the following guidelines to improve the security of your system:

- Set up official security procedures and have everyone in the newsroom follow them.
- Keep track of your backup tapes. You can get user passwords from a backup tape.
- Assign users superuser status only when they need it.
- If any user does not need superuser status, remove it. Create two user accounts for staff members who need superuser privileges—one to be a superuser and another to be a regular user. That way, you can track activity of superuser accounts.
- Change the superuser password regularly.
- Use a security modem for some of your system's dial-up connections. A security modem provides an additional layer of password protection. It offers a dial-back option, which ensures that you know the phone location of people who connect to your system.



*Security modems may not work with laptops. Contact iNews Customer Support for more information about security modems.*

- Ensure that users do not use their names, station call letters, or other easily guessed words as passwords. Require everyone to include at least one non-alphabetic character such as a pound sign (#) or a number.
- Use the system profile to set a required minimum length for all passwords so no one uses a short password.
- For devices for which you have dedicated resources—that is, devices that have a one-to-one correspondence between the physical device and device number used to identify it—use device name security on workstations in specific locations and put the device ID in a group.
- Put a MODIFY-DEV field in the queue's story form if you suspect that someone has broken into a user's account. When changes are

made to stories created after the MODIFY-DEV field has been added, the system puts the device name of the workstation where changes were made in that field.

- Be familiar with valid user accounts on your system, and the modem telephone numbers connected to your system. Restrict access to these account names and numbers to protect against unauthorized outside access.
- Use caution when distributing your staff telephone lists. Ensure that only users who need to use dial-up connections know the telephone numbers for your modems. Change these numbers occasionally for additional security.

## User Passwords

Your Avstar Newsroom Computer System protects against unauthorized access by giving each authorized user a password to log into the system. Group security, described under “Using Group Security to Control System Access” on page 18-12, lets you control specific areas of your database that each user can access.

## Checking Password Status

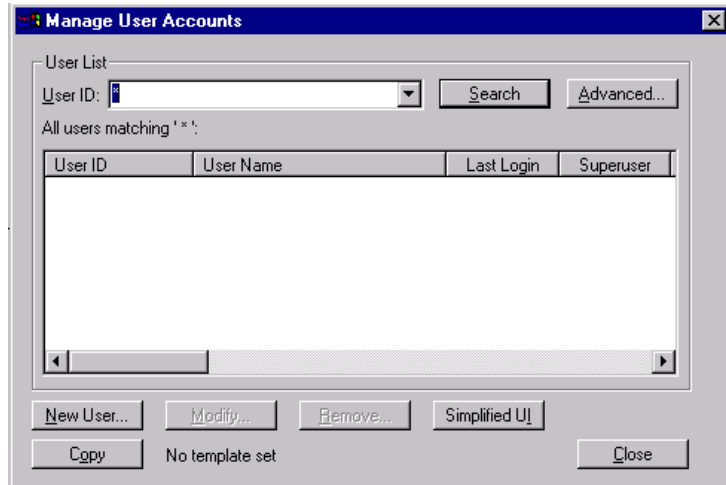
A user account without a password is an open door to your system. You should always give a user a password when you add the user to your system. See “Setting up New Users in Avstar” on page 4-22 for more information.

For information on how to check password status from the Avstar console, see “Users’ Passwords” on page G-4.

However, if you suspect that a user does not have a password or has not changed it in awhile, you can find out for certain from any Avstar Workstation. To do this, log in as a system administrator—that is, use a superuser account—and do the following:

1. Click the Tools drop-down menu.
2. Select Options.

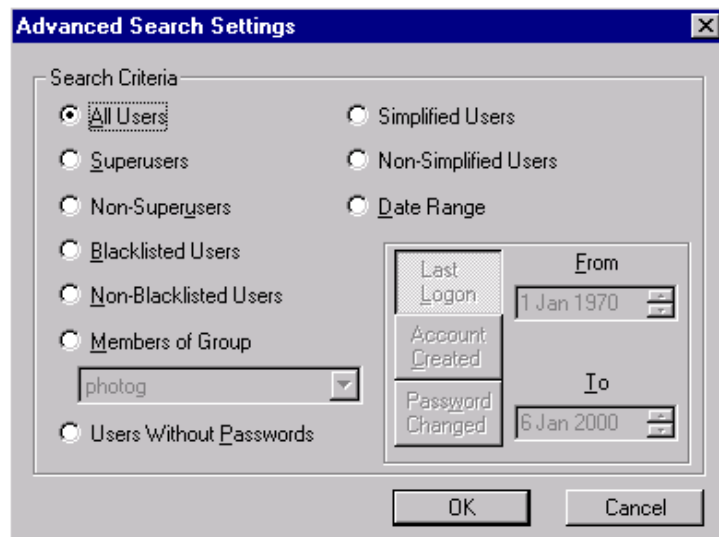
3. Select Users. The Manage User Accounts dialog box appears.



4. Type an asterisk (\*) in the User ID field if it does not already appear. Avstar recognizes this as a wildcard and therefore will search the entire database of Avstar users.

The other criteria options available in this dialog box are explained in detail in Table 4-1, “Advanced Search Criteria” on page 4-31.

5. Click the Advanced button. The Advanced Search Settings dialog box appears with All Users selected by default.





6. Do one of the following:
  - a. To search for all users without passwords, select Users Without Passwords.
  - OR-
  - b. To search for all users who have not changed their password within a specific time, select Date Range. Then click Password Changed, and specify the time frame to search.
7. Click OK to confirm your advanced search setting or click Cancel to cancel it.
8. Click Search to initiate the search.

A progress bar appears if a lengthy search is underway. Results of the search appear in the User List field in the center of the Manage User Accounts dialog box.

As a system administrator, you can change a user's password. Change a user's password to provide a new user with a temporary password or to supply an established user with a new password if the user forgets it and cannot log in. See "Changing a User's Password" on page 4-9 for more information on how to do this from any Avstar Workstation. For steps on how to change a user's password from the Avstar console using the `traits` console command, see "Users' Passwords" on page G-4.

## Forcing Individual Users to Change Their Passwords

Occasionally, you may have individual users who do not change their passwords as required. When that happens, you can force them to change their password at their next login. You can do this for a single user at an Avstar Workstation.

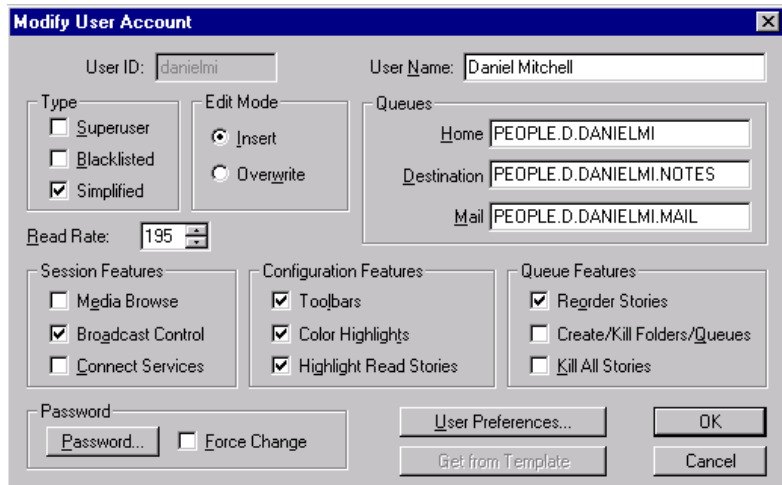
To force multiple users to change their passwords—such as all users who haven't changed their passwords in the past six months—you must go to the Avstar console. For steps on how to force password changes from the Avstar console, see “Users' Passwords” on page G-4.

### . . . At an Avstar Workstation

To force an individual user to change his or her password, do the following:

1. Click the Tools drop-down menu.
2. Select Options.
3. Select Users. The Manage User Accounts dialog box appears.
4. Type the user name in the User ID field.
5. Click Search.
6. Select the user name when it appears in the dialog box.
7. Click Modify.

The Modify User Account dialog box appears.



8. Check the Force Change box.
9. Click OK.

The next time the user logs in, he or she will be required to choose a new password.

## Tracking User Activity

Keep a record of who uses Avstar NRCS and when they use it by following the procedures described in this section. These security measures can ensure that there is no unauthorized use of your system.

At an Avstar Workstation, you can determine:

- Last login date of one or more user accounts
- Date user accounts were created
- Users currently logged in

At the Avstar console, you can determine attempted and successful logins.

## Tracking User Login Activity and Date Created

You can search for a user account's last login and the date the user account was created from an Avstar Workstation.

### . . . At an Avstar Workstation

To search for a user's last login, do the following:

1. Click the Tools drop-down menu.
2. Select Options
3. Select Users. The Manage User Accounts dialog box appears.
4. Click Advanced. The Advanced Search Settings dialog box appears.
5. Click Date Range.
6. Do one of the following:
  - a. To search for user accounts with a last login date that matches a specified date range, click Last Login.
  - OR-
  - b. To search for user accounts created within a specified date range, click Account Created.
7. Specify a date range in the From and To fields.
8. Click OK.
9. Click Search.

The requested user name information appears.

### ... At the Avstar Console

Another command you can type at the Avstar console will give you valuable information about users:

```
list u-t [<username>]
```

This command shows the date and time a user account was created, date and time of last login, and date and time of last password change.

If you do not specify a particular user name to check, you will get a listing for all users.

## Listing Users Currently Logged in

From the Avstar Workstation (ASWS), you can see a list of all users logged in, and you can find out whether a specific user is currently logged in.

To see a list of all logged-in users, do the following at any ASWS:

1. Click the Communicate drop-down menu.
2. Select Messages.
3. Select Logged In Users.



*You need to be sending a new message to use this function. Begin a new message (to anyone) to obtain access to the menu items in this step.*

A dialog box appears with a list of users currently logged in.

4. Click OK when you are done looking at the list.




To determine whether a specific user is logged in, do the following:

1. From ASWS, select the Message bar.



2. Type the user name in the To field.
3. Move cursor to the message field, using the mouse or Tab key.

One of three symbols appears to the left of the To field, depending on what you type in the To text box:

-  If the user is currently logged in, an icon of connected cables appears to the left of the user name.
-  If the user is not currently logged in, an icon of disconnected cables appears to the left of the user name.
-  If there is no such user name in the system, a question mark appears to the left of the user name.

## Recording Logins

Keep track of successful and attempted logins to preserve system security. You can spot unauthorized users, people logging in at odd hours, or repeated attempts to guess passwords.

To monitor logins from different types of devices, do the following:

1. Change the value for the `W_LOGTYPES` token in the dictionary `/site/dict/words`. A typical definition for this token might look like this:

```
W_LOGTYPES          /D
```

Each letter in the `W_LOGTYPES` value represents a different type of device that can log in on an Avstar Newsroom Computer System.

2. To track logins by device type, add the appropriate letter to the `W_LOGTYPES` value.

Use the following letters (these are the same ones that appear in the first column of a `list s` or `list c` display for these device types).

- T      Terminals, network terminals, PCs running DOS
- D      Dialup lines

- G Avstar Workstations (ASWS)
- B Web sessions

For instance, to track logins from all devices on your system, change the `W_LOGTYPES` line to look like this:

```
W_LOGTYPES          /TDG
```

The letters can appear in any order.



*On a busy system, this can create a large amount of console activity.*

3. When a user logs in at a device of a type listed in `W_LOGTYPES`, a message similar to the following is sent to the Avstar console:

```
D21 12:04:34 login arlin
```

This message includes the device type and number, time of login, and user name.

4. A logout message similar to the following is sent to the Avstar console when the user logs out:

```
D21 12:21:52 logout arlin
```

A failed login—that is, an invalid user name or password—produces a message similar to this:

```
D21 12:04:34 attempted login arlin
```

Regardless of the `W_LOGTYPES` value, a message is *always* sent to the Avstar console when a superuser logs in or out. This message includes `(n)` if the user is a superuser. An unsuccessful login by a superuser generates a message only if the device type used is included in `W_LOGTYPES`.

Using login tracking in conjunction with console history and disk logging, you can keep accurate records of who is using your system and when and where they are connecting to it.

## Using Group Security to Control System Access

Your system is designed to be used by a wide range of people. For the system to accommodate so many diverse job roles, restrict sensitive areas of your database to authorized users. The system provides a powerful security feature that lets you restrict access to important directories and queues.



**Security ensures that only authorized people can view or make changes to important queues. It does not provide absolute privacy, because superusers and user managers can open any story while performing normal system maintenance. Warn your staff not to store personal or confidential material in the database.**

Many security features in Avstar involve establishing groups and assigning privileges and restrictions to them. Chapter 6, “Groups,” gives you complete information about using the group features in Avstar NRCS to help you maintain system security.



# CHAPTER 19

## *Database Management*

This chapter provides you with information required to manage the Avstar database, where all *business data*, such as stories, is stored.

Much Avstar *system data* is stored in configuration files. See Chapter 11, “Configuration Files,” for information about managing that data.

This chapter contains the following sections:

- Monitoring Free Space
  - Understanding Database Storage Units
  - Monitoring the Free List
  - Understanding How the System Copies Stories
- Tracking Database Space over Time
  - Using the hogs Command to Obtain Information
  - Using dbpurge and dbfree to Obtain Information
- Increasing Database Space for Immediate Use
- Maintaining the Database
  - Checking the Database for Errors
  - Cleaning the Database

## Monitoring Free Space

The Avstar Newsroom Computer System is constantly collecting wire stories and adding them to the database while the news staff adds scripts, rundowns, memos, assignment sheets, and other stories. To keep from running out of disk space, Avstar NRCS tracks old stories. As stories get old, the system purges them.

Distribution and purging of disk space is called the *database cycle*. In this cycle, wire stories are collected and stored in the database for a specific interval of time. Other news items, such as scripts and assignment sheets, are created by the news staff and also kept for a preset period of time, called the *purge interval*, which is set individually for each queue. Any story older than its queue's purge interval is purged, and its space is reclaimed for new stories.

The purge interval is a database trait, so you can set different purge intervals for queues and directories, depending on the information they hold. Setting purge intervals appropriate for stories in various queues helps keep the database from growing too large.

Once an hour, at 15 minutes after the hour, an automatic `dbpurge` program scans each queue for stories older than the queue's purge interval and moves these stories into the Dead queue.

Stories sent to the Dead queue are not erased until the system needs the space. Until the system reclaims this space, stories in the Dead queue can be read, searched, edited, copied, or printed. To retrieve a story from the Dead queue, select the story and copy it to a different queue in the database, where you have write permission.

Although you can open stories in the Dead queue, they are marked for removal and will be permanently removed when the system detects the computer is running out of storage space. Your system keeps track of the space available by examining and maintaining a list of free space on the disk. The free list is explained in "Monitoring the Free List" on page 19-3.

## Understanding Database Storage Units

Your computer's disk is divided into blocks. The database portion of the disk is divided into 1024-byte blocks. When a story is saved, the system allocates as many blocks as necessary to hold the story and then divides the story among those blocks.

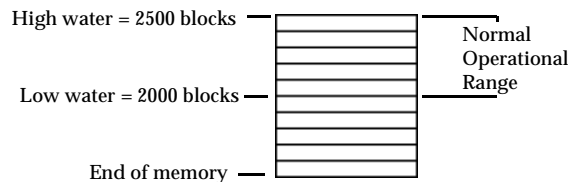
Blocks used to hold a story need not be sequential; a story can be saved in blocks that are apart. To tie together all blocks, each block contains a pointer that points to the block containing the next part of the story.

To identify which blocks currently hold file information and which are unused, the system marks blocks as either valid or invalid.

- A valid block is a block currently being used to hold some portion of a story.
- An invalid block is a block not currently being used, although it may contain information remaining from the last time the block was used. Anything held in an invalid block is overwritten the next time the system uses that block.

## Monitoring the Free List

The free list is the list of free space on the disk. By keeping track of space in the free list, the system can detect when it is running low on space. It runs an automatic `dbserver` program, which removes the oldest stories from the Dead queue and adds the space to the free list, where it is made available to the system. This way, `dbserver` maintains the volume of free space available in the database.



The free list measures space in *blocks* (a block=1024 bytes), and has a lower limit called the *low watermark*, which represents the least amount

of free space available. When the space available drops below the low watermark, the system runs `dbserver` to reclaim enough space from the Dead queue to rebuild the free list to the *high watermark*.

The `dbserver` program begins removing the oldest material from the Dead queue to rebuild the free list. Together, the high and low watermarks determine the free list's normal operational range.



*The high and low water values used by the `dbserver` program are multiples of 250—that is, 250 blocks stored in each block of the free list.*

If the system cannot get back up to the high watermark after reclaiming free space, the user will get a `low on space` message. Create free space immediately, as explained in the following section.

## Understanding How the System Copies Stories

You can configure Avstar NRCS to distribute a wire story to several queues when it is received. Likewise, two or more users can put copies of the same story into their personal queues. If you copy and distribute enough stories, a large portion of the database can become cluttered with the copies.

To avoid filling up the disk with copies of stories, your system keeps only the original story on the disk. When a story is copied to another queue, your system puts a pointer in the queue that will hold the copy. This pointer addresses the original story.

When someone opens a copy of a story, the system uses the pointer to find the original story. It makes a working copy of that story, which it sends to the user. If the user examines the working copy and makes no changes, the working copy is deleted when the user closes the story.

However, if the user makes changes to the working copy and saves it, the system saves that copy as a story, replacing the pointer.

In most cases, you can treat a pointer to an actual story as if it were the story. The only time you need to take pointers into consideration is when setting purge intervals. When the system distributes pointers to

a story in several different queues, each pointer takes on the purge interval of the queue. When a pointer becomes older than its queue's purge interval, the computer puts the pointer, not the actual story, in the Dead queue.

## Tracking Database Space over Time

Do not wait until you encounter an “out-of-space” condition before you start to think about database storage. There are two ways you can get a good picture of space usage over time:

- The “hogs” report
- The information generated by the `dbfree` command

## Using the hogs Command to Obtain Information

The `hogs` command displays how much space particular queues are using in the database. It uses this format:

```
hogs [<directory or queue name>]
```

To get a hogs report on the People directory, type:

```
hogs people
```

A screen similar to the following appears:

%	USED	SHARED	HELD	LOCKED	PURGED	QUEUE NAME
0	36	20	0	0	0	PEOPLE.LEVY.BYLINE
0	128	20	0	0	0	PEOPLE.LEVY.FINAL
0	32	40	0	0	0	PEOPLE.LEVY.FORM
...						
0	425	40	0	0	0	PEOPLE.WALTERS.NOVEL

Each line in the `USED` column contains the number of blocks used only by that directory or queue. If any directory or queue has a substantially greater number of used blocks than the others, examine that directory or queue more closely.

To obtain a hogs report on the entire database, type:

```
hogs .
```

To send a hogs report to a printer, precede the `hogs` command with `print` and the number of the printer you want to use. For instance, to send the report on the People directory to printer 3, type at the console:

```
print 3 hogs people
```

To send a hogs report to yourself, type:

```
AVSTAR-A# sh  
# hogs . | mail <your username> &
```

## Using dbpurge and dbfree to Obtain Information

To obtain an accurate idea of how much space is being used by stories in the system, do the following:

1. Empty out the Dead queue and reclaim all space used in it. Type:

```
dbserver 20000
```



*You must run `dbserver` when other programs like `dbpurge` are not running.*

2. The next day, or after completion, run the `dbfree` program to see how much space is being devoted to functions in your newsroom you consider critical.
3. Repeat this process from time to time, so you are aware of trends in space usage.

Use the information to make decisions on projected storage needs, and how space is used.

## Increasing Database Space for Immediate Use

If your database has not reached the “Low on Space” point, but you want to increase free space for immediate use, do the following:

1. Type the `dbfree` command.

```
AVSTAR-A: dbfree
```

```
data base size (2052683) free blocks (1453000) -
71% free space
```

The `dbfree` command displays database size, free list size (free blocks), and percentage of remaining database available.

2. Back up old material to tape and remove it from the database.
3. Check and reset purge intervals of Wires directories and queues.

To view purge intervals at the Avstar console, type:

```
list d wires
```

Information similar to the following appears:

```
SRPlo-LIsUG-QSXWF sortfield purge dis mbox directory
D-R-----I----Q-XW- TITLE   P4.0   D1 -   WIRES
D-R-----I----Q-XW- TITLE   P4.0   D1 -   WIRES.ALL
D-R-----I----Q-XW- TITLE   P4.0   D1 -   WIRES.AP
...
```

One of the biggest consumers of database space in a newsroom is the Wires directory. Because wire stories lose much of their value after a few days, most newsrooms set a purge interval of 2 or 3 days. In this example, all Wires directories and queues have a purge interval of four days.

The purge interval is a database trait you can customize for each database directory or queue. For more information on modifying database traits, including purge intervals, see “Changing Database Traits” on page 5-21, “Choosing Queues to be Purged” on page 5-40, and “Purge Intervals and the Purge Limit” on page 5-42.

## Maintaining the Database

The following sections describe the two most important maintenance procedures you should perform regularly on your database:

- Check the database for errors
- Clean up the database on a monthly basis

## Checking the Database for Errors

In the large databases typically maintained by newsrooms, minor errors can develop in some stories as a result of vast amounts of information the system processes every day. These errors can grow and eventually begin to damage the database if they are not removed.

To check stories in your database for errors, use the `dblines` console command, which examines every story in the database. You can run `dblines` on any server.

Since `dblines` examines every story in your database, it requires several hours to complete its task, so run `dblines` before you go home at night. By the time you return in the morning, it should be done.

When `dblines` discovers an error, it ordinarily sends an error message to the console. However, if you run it at night, have it send its messages to a printer, so they are available for you to examine.

Type the `print` command, followed by the number of the printer where you want the messages printed, `dblines`, and a period. The period tells `dblines` to check the entire database. (If you do not include it, `dblines` prompts you for a pathname.) The format for using `dblines` with `print` is:

```
print <printer number> dblines .
```

For instance, to run `dblines` on server A and have error messages sent to printer 2, select server A at the Avstar console, and type:

```
print 2 dblines .
```

When you return, check the printer for error messages. If `dblines` found any errors, it sent corresponding messages to the printer. If there are any error messages, call iNews Customer Support for assistance.



Since `print` causes `dblines` to run in the background, the console is available for normal operations while `dblines` is running. The system can display messages at the console and you can execute console commands.

## Cleaning the Database

Over the course of a month, the Avstar database may develop minor errors in its structure—the overall organization in which individual stories are arranged. These errors are the result of normal database cycles in which old material is removed and new information is added.

These errors grow and begin to damage the database if you do not remove them. The database cleanup procedure identifies errors so you can repair them before they become harmful.



*Unless you are instructed to perform this procedure more often, do a database cleanup once a month.*

There are two ways to perform the database cleanup procedure using the `dbvisit` command:

- **Offline** - This procedure requires you to briefly take your system offline, preventing users from logging in.
- **Online** - This procedure allows you to perform the cleanup while leaving your system online, except for a brief period when it will be offline to all users.



*The online procedure is not recommended because it is very complex and time consuming—that is, a procedure that takes about 30 minutes to an hour to perform offline becomes a procedure that can take several days to complete online. If you find that the offline procedure leaves your system unavailable for too long, contact iNews Customer Support for assistance with online database cleanup procedures.*

The next section explains the offline database cleanup procedure.

## Cleaning Your Database Offline

Before performing the monthly database cleanup offline, do the following:

1. Select the master computer (typically server A). See “Selecting Servers” on page 2-5 for more information.
2. Type `dbserver 20000` to empty the Dead queue.



*To check the database structure for errors, `dbvisit` must perform a cursory check of stories in the database, including those in the Dead queue. You can shorten the time it takes `dbvisit` to examine the database by using `dbserver` to empty the Dead queue first.*

*The `dbserver` command initiates a day-long operation and should be done one day ahead of the next steps in this procedure. Your system remains online during the operation.*

After the `dbserver` operation is completed, you are ready to clean up the database offline, by doing the following:

1. Shutdown the system, by completing the following steps:

- a. Take the system offline.
- b. Log out all users.

The `dbvisit` command cannot examine stories being edited. You must log out everyone on the system and stop all news programs on the servers before running `dbvisit`.

- c. Run `stop all` on *all* servers to stop all PCUs, CCUs, workstations, wires, and other devices from making further changes to the database.

After using the `stop all` console command, wait a few seconds for the prompt to reappear. When it does, proceed to the next step.



**Ensure system is offline and that you have stopped all PCUs, CCUs, and network devices. If you do not do this before cleaning your database, you may corrupt the data when you perform the cleanup.**

The procedure for shutting down the system is described in more detail in Chapter 3. Read “Shutting Down the System” on page 3-9 before performing step 3 in this procedure.

2. Start the database cleanup by doing the following:

- a. Select only the master computer (typically server A). See “Selecting Servers” on page 2-5 for more information.
- b. Become a superuser. See “Becoming a Console Superuser” on page 3-2 for more information.
- c. Type:

```
dbvisit -d
```

A message similar to the following appears:

```
Using a 40 block read buffer  
09:31:45 traversing roots
```

The `-d` instructs `dbvisit` to print a period each time it visits a new queue and a colon for every 200 stories in a queue. Once you have done a few cleanups, you can judge how far `dbvisit` has progressed by the number of periods it has printed.

When you run `dbvisit`, it begins with the root file structure and then examines the entire directory. As it inspects the root files and then the directory, `dbvisit` prints messages indicating its progress.

3. When you see the message `traversing directory`, do the following:

- a. Select the other servers on your system. See “Selecting Servers” on page 2-5 for more information.
- b. Become a superuser—if you are not one already. See “Becoming a Console Superuser” on page 3-2 for more information.
- c. Type:

```
dbvisit -dr
```

The `dbvisit` command runs in read-only mode on the other servers, but running it on all of them allows you to compare their outputs and verify the database is mirrored on each server. The `dbvisit` command continues to run on the master computer.

4. When the system prompts you with `Rebuild the free list & fix link counts (y/n)?`, do the following:

- a. Ensure `dbvisit` has completed on *all* servers (otherwise error messages may appear).
- b. Type **y** only if there are no error messages. If there are any error messages among the periods printed after `traversing directory`, contact iNews Customer Support immediately for instructions.

-OR-

- c. Type **n** to skip rebuild of the free list.

Skipping rebuild of the free list retains the old free list. While this means that unreferenced blocks are not collected back to the free list, you can add them to the free list later by performing another `dbvisit`.



**Rebuilding a free list without first fixing the errors seriously corrupts your database.**

5. If you do not see any error messages, select the master computer (typically server A) and type **y**.

In addition to spotting errors in the database structure, `dbvisit` collects any unreferenced blocks and puts them on the free list. Unreferenced blocks are not used by any story and are not a part of the free list. They are stray blocks occupying space that cannot be used until they are put on the free list.

6. Restart your system, by doing the following:
  - a. Select *all* servers. See “Selecting Servers” on page 2-5 for more information.
  - b. Exit superuser mode by pressing CTRL-D.
  - c. Type **online** to bring the system back online.
  - d. Type **restart all** to restart all devices.

The system displays `Hot-to-go` messages as the devices start. If a PCU/CCU or device cannot be started, a `failed`

message is displayed for that PCU/CCU or device. Wait for the other devices to start, then use standard procedures to restart failed devices. See “Starting the System” on page 3-5 for more information.



# CHAPTER 20

## *Backing up Your System*

To maintain your Avstar system properly, you need to perform three kinds of backups:

- *Database backups* of the Avstar database
- *Software backups* of Avstar software and UNIX operating system
- *Configuration files* like `/site/config` and `/etc/hosts`

This chapter explains the procedures you need to follow. It also gives you general information about using tapes and tape drives.

This chapter contains the following sections:

- Tape Operations
- Backing up the Avstar Database
- Restoring Data to the Avstar Database
- Disaster Recovery Planning
- Backing up Software
- Backing up Site Files

## Tape Operations

When you are making backup tapes, pay attention to the write-protect tab on the tape. If you have trouble writing to the tape, check the tab and try moving it in the opposite direction.

When loading a tape, ensure the tape has finished loading before issuing tape commands. Trying to access the tape before it is ready can result in a “hung” process that waits forever for the tape to be ready. The only solution is a reboot. For DAT tapes, wait at least a minute after all tape activity lights cease blinking to be sure the tape is properly loaded.



*Always clearly label tapes with the type of backup tape it is, date it was produced, and command used to produce it. Ensure that your tapes can be quickly found in case of emergency.*

Since the database is mirrored on all servers in your system, it does not matter on which server you produce the database backup tape. You may want to rotate which server does the database tape dump so the tape drives wear evenly and you extend the life of the drives. Alternatively, you may want to always run the backup on one server’s drive so you have another, relatively unused drive standing by in case the heavily-used drive develops problems. However, for software and site file backups, separate backups should be made on each Avstar Server.

Later in this chapter, various procedures are provided for checking data on tapes, searching for specific data on tapes, and restoring data from tapes. For more information, see “Listing Tape Contents and Backup Dates” on page 20-11 and “Searching a Tape” on page 20-14.

## Establishing Policies for Backup Procedures

Since your software and configuration files change only infrequently, you do not need to back them up very often. Your database, however,



changes hundreds or even thousands of times every day, so you need to back up much more frequently.

How often you do backups of all three types of data is up to you and your station's policies.

Here are some general guidelines:

- Since the database is mirrored across multiple servers you have built-in hardware redundancy, though please keep in mind that backups also provide insurance for rare cases of database corruption.
- One approach is weekly backups rotating through 5 tapes, one for each week of the month. If you need to keep old data for a period of time you may supplement the five tape rotation with 12 more tapes labeled with the months in the year.
- Do not store archived material beneath daily show production queues. Segregate archive material under its own directory.
- If you archive old shows on your system, you may want to produce `dbdump` tapes that cover old years, say `ARCHIVE . 1999` through `ARCHIVE . 1987`, and store them separately.
- It is a good idea to make extra copies of the tapes and store them off site. If your computer room is destroyed, you can buy new servers. But if the computer room is destroyed with all backup tapes stored in the room, your data is lost. Many corporations rent a safe deposit box for off-site storage of backup tapes.

## Backing up the Avstar Database

Database backups provide insurance against system calamities. Also, to free up space in the database, you can back up stories to a tape and then remove them from the database. You should make frequent backups of important material and the entire database. To back up database items to tape, use the `dbdump` console command.

## The dbdump Command

The `dbdump` command backs up the news database, including the People files, show rundowns, wire stories, and root database information such as user accounts, passwords, and directory traits. All user, database, and group traits are stored within the database.

The simplest form of the `dbdump` command is:

```
AVSTAR-A: dbdump c
```

The `c` stands for create. When you do a `dbdump c`, it overwrites any information currently on the tape. The `dbdump c` does a full database backup of everything (except the contents of any queue or directory with its skip flag enabled.) The *skip flag* is a database trait that prevents the `dbdump c` command from backing up contents of queues and directories with information you do not want to include in the database backup. For instance, the Dead queue and the Wires directory usually have their skip flags enabled.



*The contents of queues that have their skip flags set are not backed up during a `dbdump`. Generally, Wires queues and the Dead queue are usually skip-flagged so they are not backed up. This results in less time for the backup and less tape used. If a `dbdump` reaches the end of the tape and still has more data to back up, you will be prompted for another tape. For more information on skip flags—also known as Skip Backup database trait—see “Maintain Tab” on page 5-38.*

If you already have `dbdump` material on tape and you want to append another `dbdump` to the end of it, use the command:

```
AVSTAR-A: dbdump ad
```

An append does not overwrite `dbdump` information currently on the tape. For information on backing up individual queues, see “Backing up Individual Queues” on page 20-7.



*To minimize the impact of any potential problems, perform `dbdump` at less critical times.*

## Backing up Entire Database

To back up the entire database to tape, do the following:

1. Insert a tape into a server's tape drive.
2. Select the server that has the backup tape. See "Selecting Servers" on page 2-5 for more information.
3. Type `dbdump c`.
4. A verification request appears:

```
Do you really want to create a new archive? (n/y)
```



**When you back up the database onto a tape, anything on that tape is overwritten by the new copy of the database.**

5. Type `y` to begin copying the database to tape.

Information similar to the following appears:

```
Starting Dump(1), block(0)
dumping isam user
...
3121 stories dumped
Ending Dump(1), block(223)
```

As `dbdump` copies, the console displays messages like those above. The number of stories dumped and the ending block mentioned in these messages depends on your system.

When the console prompt returns, the backup is done. Proceed to Step 6 to verify the backup. Otherwise, remove the tape from the drive. Set the tape's write protect switch on, return it to its case, and write the date on the case label.

6. (Optional) Type `dbrestore tdv` at the Avstar console to verify the backup was complete—that everything you wanted backed up was copied to tape.

The `dbrestore tdv` console command lists every directory and queue on the tape. Since the tape contains most of the database, this list is very long.



*To list contents of a tape, the server must read the entire tape; therefore, listing contents takes approximately as long as it does to back up the database to tape.*

The output of the `dbrestore tdv` command looks like this:

```
Listing tape contents only!
Type  Stories  Name
Dir
Que           DEAD
Dir           SYSTEM
Dir           SYSTEM.KEYBOARDS
Que  1         SYSTEM.KEYBOARDS.000-INSTALLATION
...
Que  3         TEST.SMITH
3630 stories listed
```

The first column of the listing identifies whether the item is a queue or a directory. If it is a queue, the second column indicates how many stories are in that queue. The third column displays the name of the directory or queue.

If any queues in the list do not have stories listed, then either the queue has no stories, or the queue has its skip flag enabled. If the skip flag is enabled, the queue name is copied to tape, but none of its stories are copied. In the previous example, the Dead queue does not have any stories on the tape.

7. When the list is complete, remove the tape.
8. Set the tape's write protect on, and write the date on the case label.

## Backing up Individual Queues

Individual queues can be backed up by specifying the `-n` flag and listing up to 10 queue names:

```
AVSTAR-A: dbdump c -n <queuename> <queuename>
```

To back up individual queues to tape, do the following:

1. Insert tape into a server's drive and select that server.
2. Use the `dbdump` command in one of the following formats:
  - a. If you do not have a tape that already contains a backup, insert a new tape and back up the queue. For instance, to back up `SCRIPTS.2000.FEB` on a new tape, type:

```
dbdump c -n scripts.2000.feb
```

A verification request appears:

```
Do you really want to create a new archive? (n/y)
```

If there is nothing on the tape that you want to save, type `y` to continue. A message similar to the following will appear:

```
Starting track(1), block(0)
47 stories dumped
Ending track(1), block(34)
```



*When you back up a queue, `dbdump` ignores the queue's skip flag, so you can use it to back up a queue or directory that has its skip flag enabled. You can also use this process to save a queue or directory to tape before removing it from the database. For instance, if you want to restore database space by removing a queue with material no longer used, use `dbdump` to backup the queue first. Then, if you need it later, you can retrieve it from the tape.*

-OR-

- b. If you have previous backups on the tape, append the next backup to the tape with:

```
dbdump a -n <queue or directory name>
```

You can append additional backups to that tape until you use up all space on the tape. Continuing with the example in Step 2a, the next day you would insert the same tape, make sure its write protect is off, and then use `dbdump a` to append that day's scripts to the tape:

For instance, after making the first backup of `SCRIPTS.2000.FEB` in Step 2a, back up `SCRIPTS.2000.MAR` the following month by inserting the same tape and entering the following:

```
dbdump a -n scripts.2000.mar
```



*If you try to use `dbdump a` with a tape that does not already contain at least one backup, you get an `Empty` tape message. If this happens, use `dbdump c` instead.*

3. When the console prompt returns, the backup is done. Remove tape and set the write-protect switch on. Date the tape case label.
4. To verify the backup, type `dbrestore tdv`.

This list should be short, but it could take time to complete if the tape contains several weeks' worth of backups.

## Notes on Backing up the Database

If you try to back up data to a write-protected tape on the SCO platform, you get a message similar to the following:

```
SCO-A: dbdump c
Do you really want to create a new archive? (y/n) y
Starting Dump(1) Block (0), dumped on Wed Dec 13 '0:36:14: 2000

dumping isam user
Please remove tape.
Insert continuation tape (make sure safety is off)
When continuation tape is ready type: y (RETURN)
```

SGI systems return the following error message:

```
SGI-A: dbdump c
Do you really want to create a new archive? (y/n) y
write error on header - dump 1 terminated
```

If the tape is write-protected, remove it from the drive. On some tape drives you can change the write-protect without removing the tape. Set the write protect switch to off, insert the tape, and repeat dbdump.

Daily backups usually take very little time and generally do not affect system performance. However, backing up a large amount of material to tape may hinder system performance, so you should only do it during times of light system use.

If you have a very large database, the full database backup may require two or more tapes. The system prompts you to insert the next tape. Remove the current tape, number it as tape number one, and insert another tape. The server automatically continues copying the database when you insert the second tape.

## Restoring Data to the Avstar Database

The following sections describe how to restore data from tape.

### The dbrestore Command

Tapes produced with the `dbdump` command can be read back in and restored using the `dbrestore` command. The most common use of the `dbrestore` command is to restore a specific queue or directory. This is done by specifying the name on the command line:

```
AVSTAR-A: dbrestore d -n <queuename> <queuename> ...
```

You can use the `dbrestore tdv` command to verify that you can properly read a `dbdump` tape.

Check free space before you restore large amounts of data to the database. Before you `dbrestore` stories or queues to your database from tape, ensure you have enough free space available to hold the restored stories. You can create free space using the `dbserver` command to empty out some of the Dead queue and add that space to the free list. A conservative estimate would be to have 10 free blocks for every story you plan to restore.

### Restoring a First-Level Directory

Restoring a first-level directory, such as `Scripts`, as opposed to `SCRIPTS.FEB`, is the same as restoring queues and other directories. However, when you restore such a directory, you also have the option of restoring it under a new name. Use:

```
dbrestore d -n <old dir. name>[=<new dir. name>]
```

If the new directory does not already exist, `dbrestore` creates it.

For instance, suppose you have a directory called `Scripts` and you want to restore a previous backup of this directory to another directory called `Scripts-temp`, which does not exist. Type:

```
dbrestore d -n scripts=scripts-temp
```



A message similar to the following appears:

```
744 stories restored
```

Since `Scripts-temp` does not exist, `dbrestore` creates it before restoring `Scripts`. When `dbrestore` has restored all stories in `Scripts`, it displays a message indicating how many stories were restored.



*If you want to restore a specific directory, be sure to always include the `-n` and the directory name. If you do not, `dbrestore` tries to restore everything on the tape to your database, which could cause your system to run out of space.*

You cannot restore a first-level directory to a lower level. In the example above, for instance, we could not have restored the `Scripts` directory to `SCRIPTS.MAY`.

When you restore a first-level directory to an existing directory, it acquires the traits of that directory. If `dbrestore` must create the directory, it uses database traits of the directory it is restoring.

## Listing Tape Contents and Backup Dates

When restoring a directory, you may need to examine the tape's contents to ensure the directory you want to restore is actually on that tape. Otherwise, if you attempt to restore a directory or queue that is not on the tape, the system may take several minutes to search the entire tape before reporting `no stories restored`.

The pages that follow show you how to use the `dbrestore` command to list the directories and queues that have been backed up on a tape, to find out if the tape contains any stories saved after a certain date, and to find out the date of each backup on the tape.

## Listing Contents of a Tape

To list every directory and queue on the tape, as well as the number of stories in each queue, to the server with the tape drive, type:

```
dbrestore tdv
```

Information similar to the following appears:

```
Listing tape contents only!
Type  Stories  Name

Dir
Que           DEAD
Dir           SYSTEM
Dir           SYSTEM.KEYBOARDS
...
Que  3         TEST.SMITH
3630 stories listed
```

Each line in the list consists of three columns and identifies a directory or a queue that has been saved to the tape:

- The `Type` column indicates whether the line lists information about a directory or a queue. Directories are identified by `Dir` and queues by `Que`.
- The `Stories` column is used only in lines beginning with `Que` and identifies how many stories are in that queue. Use this number as a guide to how much space you need to restore the queue. If there is no number, the queue is empty. (The queue is empty because either no stories existed in the queue or the queue's skip flag was enabled.)
- The `Name` column identifies the directory or queue by name.

If you attempt to restore a queue with no stories, you see the following message:

```
no stories restored
```

## Listing Items Dumped on a Particular Date

To list items dumped on a particular date:

1. Use this format of `dbrestore`:

```
dbrestore td -d <date>
```

2. Enter the date in `YYMMDD` (year, month, day) format.

Begin single-digit months and days with a zero, such as `04` for April. Also check for items dumped to tape during a month by using the year and the month (`YYMM`).

You can check to see what was dumped to a tape during a year by specifying just the year (`YY`).

For instance, to list everything dumped in October 2000, type:

```
dbrestore td -d 0010
```

A message similar to the following appears:

```
Listing tape contents only!
Type  Stories  Name
Dir
Dir          SHOW.TODAY
Que   34      SHOW.TODAY.SCRIPTS
Que    3      SHOW.TODAY.RUNDOWNS
Continuing with Dump(2) Block(0)
Dumped Thu Oct 19 10:30:19 2000
Dir          SHOW.TODAY
Que   27      SHOW.TODAY.SCRIPTS
Que    3      SHOW.TODAY.RUNDOWNS
67 stories listed
```

In the example, the tape contains only backups from two dates: the first backup and one made on October 19, 2000. Notice the date of the first backup is not included in the listing. To find the date of the first backup, use the `dbrestore tf` command described below.

The `dbrestore` displays a `Continuing with Dump` message just before processing the second dump.

## Listing the Date of Each Backup

To list the dates of each backup on a tape, type:

```
dbrestore tf
```

A message similar to the following appears:

```
Listing tape contents only!
Starting:  Dump(1)  Block(0)
Dumped Tue Oct 10 15:33:01 2000
Continuing with Dump(2)  Block(0)
Dumped Thu Oct 19 10:30:19 2000
Elapsed time: 00:13
```

In the example, the tape contains two backups, one done on October 10, 2000, and another done on October 19, 2000.

## Searching a Tape

You can retrieve individual stories from a tape by searching for a word contained in the story.

### The `searchtape` Command

The `searchtape` command allows you to search through a tape looking for a specific word. Stories that contain that word are restored to the `SYSTEM.SEARCHTAPE` queue. There is a maximum number of stories or hits that will be restored with the `searchtape` command. This default is specified in the `/site/system` file. See “Searching a Tape by Word(s)” on page 20-15 for more information.



*When you list a tape’s contents, your system must read all information on the tape to generate a list of the contents. If the tape contains much information, listing the tape’s contents takes a significant amount of time.*

*Because searching tapes for stories takes a long time and can reduce system performance, restrict tape searches to periods of light system use. It is recommended you run all tape operations during non-critical periods, not during shows.*

## Searching a Tape for Stories

To search a tape for stories, do the following:

1. Combine one or more keywords with a date or range of dates, as described in the following sections.
2. When the system finds a story matching the search criteria, it restores the story to the database. Because stories on tape may be old versions of stories still in the database, the system restores them to the `SYSTEM.SEARCHTAPE` queue. This prevents the system from putting an old version of a story in a queue that contains the current version of the same story.
3. After you restore a story to `SYSTEM.SEARCHTAPE`, you can move it to any queue.



*Before restoring stories from tape, check how much free space remains in your database at the Avstar console. See “Checking for Free Space on a Database” on page 20-18 for more information.*

## Searching a Tape by Word(s)

To search a tape by word(s), follow the `searchtape` command with the word `for` and the word(s) for which you want to search.

Use this form of the command:

```
searchtape for <words>
```

If you include more than one search word, the system searches for stories that contain any words for which you are searching. You can search for up to 20 words at a time with this command.

For instance, to search for stories that contain either helicopter or president, type:

```
searchtape for helicopter and president
```

A message similar to the following appears:

```
8 stories restored to SYSTEM.SEARCHTAPE
```

This command finds stories with the word `president` and it also finds stories with the word `helicopter`. It would not limit the results to stories that contained *both* `president` and `helicopter`.



*It is not possible to construct Boolean (and) searches using the `searchtape` command.*

### Searching a Tape by Word and Date Range

To restrict a search to stories backed up between certain dates, do the following:

1. Include the range of dates to which you want to restrict the search in the `searchtape` command.

The command format for this kind of search is:

```
searchtape from <date> to <date> for <words>
```

Dates must be in YYMMDD (year, month, day) format, and single digit months and days must begin with a zero (for instance, April is 04). If you want to specify only a month or year, shorten the date format to YYMM or just YY.

2. Follow the second date with the word `for` and the word(s) for which you want to search.

To restore stories archived between May 1, 2000, and May 21, 2000, that contain the word `helicopter`, type:

```
searchtape from 000501 to 000521 for helicopter
```

A message similar to the following appears:

```
6 stories restored to SYSTEM.SEARCHTAPE
```



*`searchtape` looks for stories saved to tape within the dates you specify. It does not look for stories created within those dates.*

### Searching a Tape by Word and Day

To search for stories saved to tape on a particular day, use the search format:

```
searchtape just <date> for <words>
```

To search for stories backed up on May 15, 2000 that contains the words helicopter or president, type:

```
searchtape just 000515 for helicopter president
```

### Searching a Tape by Word and Month

To search for stories saved to tape during a particular month, follow `searchtape` with `just` and shorten the date to the year and month (YYMM).

For instance, to search a tape for any story saved to tape in February, 2000 that contains helicopter, type:

```
searchtape just 0002 for helicopter
```

### Specifying a Maximum Number of Stories to Search

To specify a maximum number of stories for `searchtape` to restore, include the word `max` and the number of stories you want to restore.

To search a tape for all stories that contain the word `president`, type:

```
searchtape for president
```

If you do not specify a maximum, the system restores up to 50 stories.

If you specify a maximum number, `searchtape` stops when that number of stories has been restored, or when all stories containing the text have been found—whichever comes first.

To search a tape for the first 10 stories containing `president`, type:

```
searchtape max 10 for president
```

Anytime you specify more than 50 stories to restore, you are reminded that this could cause the system to run out of disk space.

When the search is complete, `searchtape` stops and displays a message indicating the number of stories it has restored.

To cancel the restore operation at any time while in progress, press Delete.

## Checking for Free Space on a Database

To check free space on a database, do the following:

1. Select a server. See “Selecting Servers” on page 2-5 for more information.
2. Type:

**dbfree**

A message similar to the following appears:

```
data base size (44598) free blocks (2500)
```

You should have 10 free blocks for each story you want to restore. For instance, if you expect to find 10 stories to restore, you should have 100 free blocks. Add an extra space to the free list before conducting a search because the search may find and restore more stories than you expect.

## Adding Blocks to the Free List

To add 100 blocks to the free list, do the following:

1. Select the master computer (typically server A). See “Selecting Servers” on page 2-5 for more information.
2. Type:

**dbserver 26**



The number after `dbserver` should be the sum of the additional blocks and the free blocks from `dbfree` divided by 100.

A message similar to the following appears:

```
AVSTAR-A Sat Dec 23 10:29:26 2000 dbserver complete-exiting
```

At this message, `dbserver` has finished and you can proceed. You can run `dbfree` again to verify that you have enough space.

## Notes on Restoring the Database

When a story being restored has more than 1,000 lines, the following message is displayed:

```
Warning: Large story being restored.  
Queue: FUTURES.FEBRUARY  
Title: "Senate Hearing"  
Size: 1004 lines (approximate)  
Press <del> to skip this story.
```

If the story size continues to grow, the following message prints every 1500 lines:

```
Large story being restored - lines 1500 title "Senate Hearing"  
Press <del> to skip this story.
```

To skip the large story, press Delete while it is being restored.

Always include `for` and one or more words for which you want to search when you use `searchtape`. Even if you use a date or a date range as part of the search criteria, you must include a word or words for which you want the system to search.

Use search criteria as specific as possible:

- Always use a narrow range of dates

- Use a small number of search words
- Use words likely to appear only in stories for which you are searching

## Disaster Recovery Planning

Having a complete `dbdump` of the database ensures that you can restore your data in case of calamity; however, restoring a complete `dbdump` to a freshly initialized database can be very time-consuming and leave the newsroom without access to the system until the full database `dbrestore` finishes. On very large databases, the full database `dbrestore` can take from 12 to 36 hours.

Planning a separate, efficient disaster recovery `dbdump/dbrestore` procedure can significantly speed up the time before the system becomes functional to users.

## Disaster Recovery Dbdump

The first step is to create a stripped down `dbdump` that contains the directory structure and minimal amount of queues and directories needed to bring the system online. After the minimal `dbdump` was restored, users can start working while the remainder of the database is restored from the last full database `dbdump` tape.

### Create Minimal `dbdump`

The entire structure of the database, along with your designated minimal set of queues, can be backed up, specifying the list of essential queues to be backed up:

```
AVSTAR-A# dbdump c -n <queuename> <queuename> ...
```

The essential directories would be System and possibly Show and a few daily Assignments queues. The System queue *must* be a part of the minimal dbdump.

If you store your master rundown skeletons elsewhere, add them to your list of essential queues. This may be the bare minimum of directories needed to get a show on the air.



*Do not store archives of shows beneath the Shows directory. Having years of archive material mixed in along with your rundowns and master rundowns will slow the dbrestore significantly. If restoring Assignments queues, ensure you restore only daily queues needed to operate; do not include all future files in the minimal dbdump.*

You can list up to 10 directories and queues with the `-n` option of the dbdump program.

```
AVSTAR-A# dbdump c -n system shows assignments
```

In case of disaster, this minimal dbdump would be the first tape restored to the system. Since it will not contain many stories, to dbdump to or dbrestore from this tape should be relatively quick and users can resume work while the remainder of the material trickles in from the full dbdump tape.

This minimal dbdump tape should be updated periodically.

## Disaster Recovery Dbrestore

If a disaster occurs, restoring the entire database from these disaster recovery dbdumps should only be done under the guidance of iNews Customer Support.

The steps are:

1. Initialize database.
2. Restore minimal dbdump.

3. Restore from latest `dbdump` tape.
4. Restore separate archive material tape (if separately `dbdumped`).

The system could be brought up and made available for users shortly after the initiation of step 3. Most queues on the system would be empty at first, but would gradually fill in as the latest `dbdump` tape was restored.

Some cleanup must be done to eliminate duplicate stories in the essential queues, since material was restored to these queues from two separate tapes, but this would be a minor trade-off for this quicker method of getting the system back up and users online.

## Backing up Software

Your system uses programs and site-dependent configuration files, called site files, to adapt it to your newsroom operations and run the workstations, printers, wires, and other devices. These programs and site files make your system software unique from that in any other newsroom.

It is important you keep a current copy of your software on tape in case you need to restore the software area of the disk. Anytime you make major changes to the system software—such as extracting new programs from an update tape—make a new software backup.

Likewise, when you make major changes to a site file, update the software backup to include these changes. For instance, if you add a new workstation to your system, you define that workstation in the configuration file. This represents an important change to this site file so, after modifying this file, update your software backup tape.



*If your system uses the SCO UNIX or SGI Irix operating system installed on it, you need to make a separate `softdump` tape for each server in the system. (Each tape contains information unique to that server.)*

To make a separate `softdump` tape for each server, repeat the following procedure for each server in your system. After backing up a server's software, be sure to label the tape to indicate which server's software is on it.

You should do a `softdump` whenever your software is updated, if you add software, or when an operating system patch is applied.

## The `softdump` Command

The `softdump` command backs up the entire software area to tape. This includes the UNIX operating system as well as the Avstar software. A `softdump` tape allows you to restore the system to disk in case of accidental deletion or hardware failure.

You must be a superuser to run `softdump`:

```
AVSTAR-A# softdump
```

## Using the `softdump` Command to Back up System Software

To back up your system's software to tape, do the following:

1. Check the software backup tape and ensure that the write-protect switch is off.
2. Insert the tape into the master computer (typically server A).
3. Perform the backup using the `softdump` command.
  - a. Select the master computer. See "Selecting Servers" on page 2-5 for more information.
  - b. Become a console superuser. See "Becoming a Console Superuser" on page 3-2 for more information.
  - c. Type:

```
softdump
```

The `softdump` program takes several minutes to copy your software. It may print several messages. Among these, it may print `read: I/O error`. Ignore this message, because it is the normal result of reaching the end of one part of the disk.

4. When the `softdump` is done and the system prompt returns, press CTRL-D to exit superuser mode.
5. Remove and label the tape. Include the date and server name.

## SCO Emergency Boot and Root Floppies

Besides the `softdump`, `dbdump`, and `sitedump` tapes, SCO UNIX systems may also require emergency boot and root floppy diskettes to restore software in case of an emergency. A set of emergency boot and root floppies should have been made when your system was installed.

You can create emergency boot and root floppies by doing the following:

1. Become superuser.
2. Enter the SCO `mkdev fd` command.
3. Follow prompts to create diskettes.

---

## Backing up Site Files

Your system uses site files in combination with software taken from your release and update tapes. When you do a software backup, both your system software and site files are copied to tape.

If something happens to your software backup, you may recover your programs from the release tape, but your site files could be lost. In addition to backing up the software, back up your site files on a separate tape. If anything happens to the software backup, you can use your release tape and site file backup to rebuild your system software.

Make a new site file backup after making significant changes to any site file. For instance, adding a printer to your system requires that you modify your configuration file (`/site/config`) and create a new printer profile. After making these changes, make a new software backup and then make a new site file backup.

To make a separate `sitedump` tape for each server, repeat this procedure on each server. After backing up a server's site files, ensure you label the tape to clearly indicate which server's site files are on it.

## The `sitedump` Command

The `sitedump` command backs up files in the `/site` directory and certain key files in the `/etc` directory. This is where all of your site-specific configuration files are kept. The tape should be updated whenever you make changes to your configuration.

To back up your system's site files to tape, do the following:

1. Insert the tape into the master computer (typically server A).
2. Select the master computer. See "Selecting Servers" on page 2-5 for more information.
3. Become a console superuser. See "Becoming a Console Superuser" on page 3-2 for more information.

4. Type:

**sitedump.**

This takes a few minutes, and no messages are displayed.

SCO Systems

SCO UNIX systems can make sitedumps to their floppy drives. The floppy diskette does not need to be formatted first. Insert a 1.44 MB diskette in the system, and type:

**sitedump -f /dev/fd0**

5. When the `sitedump` is done and the prompt returns, exit from superuser mode by pressing CTRL-D.
6. Remove the tape and label it. Include the date and type of backup.



# CHAPTER 21

## *Disconnects*

This chapter provides information to help you recover from a system disconnect. The sections include:

- Normal System Status
- Detecting a Disconnect
- Types of Disconnects
- Causes of Disconnects
- Disconnect Recovery
  - Overview
  - Procedures
  - Worksheet

## Normal System Status

On a dual server system, the status command will show the system is AB, with both servers connected to each other. A display similar to the following will appear on server A:

```
AVSTAR-A# status
A is ONLINE and has been CONFIGURED. ID is AVSTAR
System is AB. Master is A.
Disk status is OK
```

The system status is reported identically on both servers in the system. For instance, a display similar to the following will appear on server B:

```
AVSTAR-B# status
B is ONLINE and has been CONFIGURED. ID is AVSTAR
System is AB. Master is A.
Disk status is OK
```

When the system is dual and databases are mirroring between the two servers, a story written on one server is automatically mirrored to the database of the other server. When the system is connected and has the normal prompt, the servers are in communication with each other and the disk mirroring process is active.

## Detecting a Disconnect

If the servers disconnect from each other, the link is severed and stories written on one are no longer mirrored to the other server. If the servers disconnect, a disconnect warning message appears on the Avstar console.

On SCO UNIX Systems

On the SCO UNIX (Intel) platform, the messages include detailed information from the driver (mp) that controls the mirroring:

```
B Sat Dec 30 07:24:29 2000 bioserver lost TCP connection from A,
readcount=0 (115)
mp: A not responding
mp: still trying
mp: rpack F0365C04 seq 13750 preferred 00000001 son 00000001
failarg 00000000
mp: startfail 00000000 sdead 00000000 first 00000002 nactive 1
access 1
mp: nompsbufs 0 utotal 1 nactive 0 rpacks 487154 rexmits 42
mp: noumatch 0 numreq 473406 xpacks 487198 nosenddata 0
mp: request (15 00000000 0 8 4)
mp: pack (2 66 953723 11674 13749)(0)[9584]
mp: bactive 953723 0
mp: disconnecting
mp: rpack F0365C04 seq 13750 preferred 00000001 son 00000001
failarg 00000000
mp: startfail 00000000 sdead 00000001 first 00000002 nactive 1
access 1
mp: nompsbufs 0 utotal 1 nactive 0 rpacks 487154 rexmits 45
mp: noumatch 0 numreq 473406 xpacks 487201 nosenddata 0
mp: request (15 00000000 0 0 8)
mp: pack (2 66 953723 11674 13749)(0)[9584]
mp: bactive 953723 0

B Sat Dec 30 07:26:05 2000 db : COMPUTER LINK FAILED

B Sat Dec 30 07:26:06 2000 bioserver disconnecting

B Sat Dec 30 07:26:06 2000 disconnect A COMPUTER DISCONNECTED
```

On SGI Systems

On the SGI platform, messages are slightly different and contain less detailed information, as follows:

```
A Mon Jan 3 12:19:03 2000 WD manager B silent for 30 seconds
A Mon Jan 3 12:19:03 2000 RCO manager : LINK TO B FAILED,
DISCONNECTING B
A Mon Jan 3 12:19:11 2000 attach 66 to server on computer A
S201: 12:19:29 Hot-to-go
S202: 12:19:29 Hot-to-go
S203: 12:19:29 Hot-to-go
S204: 12:19:29 Hot-to-go
S205: 12:19:29 Hot-to-go

A Mon Jan 3 12:19:33 2000 disconnect B COMPUTER DISCONNECTED
```

Similar warning messages appear on the Avstar console for other servers in the system.

In addition to console warning messages, a warning message is broadcast to all users currently logged in to the system. This warning appears on the status bar (if displayed) at the bottom of the Avstar session.

After the system has disconnected, the databases are no longer mirrored between the two servers and immediate attention is required. Since mirroring has stopped, stories written by users on the A server will not be seen by users logged in to the B server. There will be two entirely separate databases.

There is no way to reintegrate two disparate databases. You must decide which database will be retained as the master database, and the other database must be erased. The steps for recovering from a disconnect are outlined later in “Procedures” on page 21-8.

When users note the disconnect message from the system, they may want to export the story they are working on to a file, in addition to normally saving it to the server. They can export the story to a file on their local hard drive by clicking the File drop-down menu and select-

ing Export Story. This will make a backup copy of the story on the local drive of their PC. Since they may be logged in to the server containing the non-master database that will be erased, this will give them a backup copy, which could be re-imported if necessary.



**When the servers disconnect, stories written on server A are not mirrored to server B and stories saved on B are not mirrored to A. Users should be alert to the disconnect warning and contact their system administrator immediately!**

You can check to see if the servers are connected to each other and mirroring at any time using the `status` command. If they are connected and properly mirroring, they will both agree on the system status and it will report that the system is dual (AB):

```
System is AB.
```

## Types of Disconnects

When the servers disconnect, one may disconnect from the other, or they may both disconnect from each other.

If they have disconnected from each other, each will report that it is a single system, with itself as the master:

```
AVSTAR-A# status
A is ONLINE and has been CONFIGURED. ID is AVSTAR
System is A. Master is A.
Disk status is OK
```

The single system status is reported identically on both servers:

```
AVSTAR-B# status
B is ONLINE and has been CONFIGURED. ID is AVSTAR
System is B. Master is B.
Disk status is OK
```

The other possibility is that servers will disconnect, but one of the servers will not note the disconnect. In this case, one will report that the system is single while the other states the system is still AB.

Regardless of the report, once one of the servers has disconnected, the system must be recovered following the procedures in this chapter. The steps outlined are the only way to recover and get the servers back in mirror.



**If the system has disconnected, you cannot simply reboot the servers and bring it back up normally. Rebooting and connecting two servers together after a disconnect can lead to database corruption!**

## Causes of Disconnects

Servers are normally in constant communication with each other. When a story is saved, the server tries to mirror that change across to the other server's database. If the server cannot contact the other server for a period of 30 seconds, it assumes the worst—that the other server has died and is not available and that as the surviving server it must be responsible for the entire system.

Knowing this design, it is obvious that network outages will cause a disconnect, as will the loss of power by one server.

A “dirty” network leading to numerous network output errors (called `Oerrs`, as revealed by the `netstat -i` command) can cause a disconnect, particularly if the output errors are rapidly climbing.

A software error that leads to a looping condition that causes a server to become so busy it cannot respond to a mirroring request could also theoretically lead to a disconnect.

Hardware failures such as the failure of a network card or hard drive may also lead to disconnects. Contact iNews Customer Support for help if you lose a hard drive.

## Disconnect Recovery

This section provides an overview of recovering your system from a disconnect, recovery procedures, and a quick reference worksheet you can use should a disconnect occur.

### Overview

Once a system has disconnected, one server must be selected to continue on as the master computer. This server will be referred to as the *survivor*. The other server will become the *failed server*.

Before the failed server can be reconnected to the survivor, it must be rebooted and its database wiped clean. Once the database on the failed server has been cleared, the server can be reconnected to the survivor and the master database copied back across from the survivor.

Because one server's database will be selected as the master database and the other's database erased, the sooner a disconnect is discovered minimizes the possibility of data loss.

In normal dual-server operation, half the devices and sessions are configured on one server and the other half are configured on the other server. The most important thing to do after a disconnect is to reconfigure the survivor so that it knows it must be responsible for ALL devices and sessions. You can then restart any network devices that were running on the failed server.

The steps are covered in more detail in the next section, "Procedures" on page 21-8.

The disconnect recovery steps are as follows:

1. Reconfigure the system (survivor).
2. Restart the failed servers's network devices (PCU/MCSPCs) on the survivor.
3. Reboot the failed server.
4. Clear the database on the failed server.

5. Reconnect the failed server.
6. Copy the database to the failed server.
7. Log out and stop the failed server's transferred network devices, sessions, and servers running on the survivor.
8. Reconfigure the system.
9. Start up the transferred network devices and servers back on the failed server.

## Procedures

This section contains an example of recovery from a disconnect of a dual AB system. In the example, it is assumed that server A was chosen as the survivor and server B was designated the failed server:

AVSTAR-A:            survivor

AVSTAR-B:            failed server

The failed server is also referred to as the *revived server* once it is reconnected to the system.



*If you reverse the roles of the servers during your disconnect and make B the survivor and server A the failed server, remember to adjust which server you issue the commands on appropriately! The steps shown in the following example assume server A is the survivor; reverse A and B throughout the process if server B is chosen as your survivor.*

It is assumed (for this example) that when the servers disconnected, server A became a single system (`system is A, master is A`) and server B also became a single system (`system is B, master is B`). In this situation, users logged in on A can continue working and saving stories. Users logged in on B can also continue working and saving stories, but since the servers are not mirroring, anything saved to server A's database is not copied over to server B's database, and vice versa.



The databases on the machines are no longer mirrored. The only recourse is to choose one of the servers' databases to become the master database. The database on the other server is wiped clean and then recopied from the master server.

To export a story to a local hard drive, click File drop-down menu and select Export Story.

Users logged on to the failed server (B) are creating stories on a database that is going to be wiped clean. That information will be lost unless stories created after the disconnect are first exported to the local hard drive so they can be imported to the survivor once the user logs in to the other server.

The following are steps necessary to recover the system:

### **Step 1: Choose one master machine to continue working with (survivor)**

It does not matter whether you choose server A or B as your master computer. What is important is to choose one quickly. You may choose one server over the other if it has more users logged in on it. Or you may choose the server that has the show producer logged in on it. If you are about to enter a show, you may pick the one that runs the teleprompter. Or you may choose A as the survivor so the steps exactly shadow instructions in this section.

### **Step 2: Reconfigure the master computer (survivor)**

When you run the `configure` command, the master computer looks at the current system configuration and then consults the appropriate host section of the configuration file (`/site/config`) to see devices for which it is responsible.

If the system is A, then the configuration section for host `a a` is consulted. This section will contain all the device and session numbers, including those that normally run on the failed server (B).

For more information, see “Becoming a Console Superuser” on page 3-2, “Selecting Servers” on page 2-5, and “Shutting Down the System” on page 3-9.

**You must be a superuser to configure the system, so become superuser:**

```
AVSTAR-A: su
Password:
AVSTAR-A#
```

**and then run this command sequence:**

```
AVSTAR-A# offline
AVSTAR-A# configure
AVSTAR-A# online
```

**Step 3: Halt and reboot the disconnected (failed) server.**

As a superuser, type the following commands:

```
AVSTAR-B# sync
AVSTAR-B# init 6
```

**Step 4: Reboot the disconnected (failed) server’s network devices (CCU/PCUs and MCSPCs).**

It may not be necessary in all cases to reboot the CCU/PCUs and MCSPCs. They may restart normally without a reboot. Reinitializing them by cycling the power virtually guarantees successful restarts.



*Keep a current printout of your /site/config file near your control console for handy reference during emergencies such as disconnects.*

**Step 5: Restart the disconnected (failed) server’s network devices (CCU/PCU/MCSPCs) on the survivor.**

Restart the failed server’s CCUs and MCSPCs on the survivor:

```
AVSTAR-A# restart <number>
```

You may string multiple device numbers on one line, like this:

```
restart 10 20 30 19 145
```

Once you restart all failed (disconnected) server's network devices on the survivor, the survivor will be running all devices in the system. It is not necessary to restart the failed server's utility programs, because they are automatically started on the survivor when the disconnect is detected.

**Step 6: Log in to the rebooted failed server as system operator, then become superuser.**

See "Logging in As System Operator" on page 3-2 for more information.

At the login prompt, log in as the system operator user (so):

```
Avstar News System
login: so
Password:
IRIX Release 6.5 IP32 avstar-b
Copyright 1987-1998 Silicon Graphics, Inc. All
Rights Reserved.
Last login: Mon Jan  3 10:07:11 EST 2000 on ttyq1
?:
```

See "Becoming a Console Superuser" on page 3-2 for more information.

When you log back in you will be at the question mark prompt because the server is not yet named (connected). You must be superuser for the next step, so become a superuser:

```
?:su
Password:
?#
```

### Step 7: Wipe clean the disconnected (failed) server's database

Select the failed (disconnected) server on the console and use the `diskclear` command to wipe the database off the failed server. The display will look similar to the following (with what you type appearing in bold):

```
?# diskclear -
DANGER -- This program DESTROYS all information on
this computer's data base.
Do you wish to save the current data base? (y/n): n
Are you sure you wish to CLEAR the disk? (n/y): y

? Mon Jan  3 16:18:23 2000 diskclear CLEARING DATA
BASE

.....10.....20.....30.....40.....5
0
.....60.....70.....80.....90.....
100
.....110.....
```

The `diskclear` will print a sequence of dots and numbers as it clears the disk. The `diskclear` may take 20-40 minutes to complete.

### Step 8: Reconnect the failed server to the survivor.

See "Selecting Servers" on page 2-5 for more information.

After the `diskclear` has completed you can reconnect the servers. Select both (*all*) servers on the console and type:

```
reconnect <failed> master=<survivor> net=ab
```

Depending on which one failed and which one is master, you would enter one of the following commands:

```
reconnect a master=b net=ab
```

-OR-

```
reconnect b master=a net=ab
```

A few moments after entering the command, the failed server will regain its normal, named prompt. Communication and mirroring between the servers is reestablished.

**Step 9: Begin copying the database back over to the failed server.**

Select the revived (failed) server only on the console and begin a slow `diskcopy`:

```
AVSTAR-B# diskcopy -s
```

Alternatively, you can perform a faster `diskcopy` with the `diskcopy -` command. The slow `diskcopy` takes longer but has less impact on system performance.



*The slow `diskcopy` is recommended on SCO servers with two database partitions on the same physical disk.*

Users can continue working while the `diskcopy` continues in the background.

The following steps can be performed at any time after the reconnect.

**Step 10: Stop the revived server's network devices (CCU/PCUs and MCSPCs), utility programs (such as action servers), and sessions on the survivor.**

At this point, all devices and sessions are running on the survivor—that is, master computer—even the ones that normally run on the other, revived server. These devices, utility programs, and sessions eventually need to run in their normal place on the revived server.

Two methods are available for splitting the workload and putting things back in their normal places. The first method involves logging off and stopping half of the system. The second method is easier and involves briefly logging off all users and stopping all devices.

## Method 1

If you cannot afford to log all users off for 10 minutes or so, the first step to transferring them back to their rightful place is to determine which devices and sessions belong on the revived server. Those devices and sessions must be logged out and then stopped. Utility programs (such as action servers and txnet links) need not be logged out; they can be stopped directly.

```
AVSTAR-A# logout <number>
```

```
AVSTAR-A# stop <number>
```

Consult the appropriate host section of the `/site/config` file to see which devices must be transferred back to their normal locations. Before stopping them, take the system offline to prevent new logins, then type `list s` to see whether users are logged in on the session numbers that must be stopped. These users must log out for a few minutes. Log them out, then stop the devices and servers:

```
AVSTAR-A: offline
```

```
AVSTAR-A: list s 10
```

```
P11                                     A
```

```
P12                                     A
```

```
T15          palmer                    A
```

```
D16                                     A
```

```
AVSTAR-A: logout 10
```

```
AVSTAR-A: stop 10
```

```
AVSTAR-A: restart 10
```

Only Avstar Workstation sessions need to be logged out; they need not be restarted since users initiate the sessions when they login.

You may need to consult your printout of the `/site/config` file to determine which server and session numbers must be stopped. They are the numbers in the `host ab b` section. Ensure that you stop *all* of the failed server's utility programs—action, txnet, rxnet, and special servers—that are currently running on the survivor.

## Method 2

On larger systems, logging off and stopping half of the devices, utility programs (such as action servers), and sessions is sometimes more difficult and time consuming than simply logging everyone off and stopping all devices. Rather than hunting through the `/site/config` file and determining which sessions and devices belong to the failed server, it may be quicker to schedule a time to bump everyone off.

Alternatively, if you have the opportunity to schedule a time to log the users off and stop the devices, you can opt for the second method and instead run the following sequence of commands on the survivor:

```
AVSTAR-A#  offline
AVSTAR-A#  logout all
AVSTAR-A#  stop all
```

**Step 11: Reboot the PCUs/CCUs/MCSPCs that you have stopped.**

Rebooting the network devices before getting them restarted ensures the devices come up cleanly.

**Step 12: Reconfigure the system (survivor).**

The system is now running in a dual AB configuration. When you run the `configure` command, the system will reconsult the `/site/config` file and divide responsibility for which server will run the devices, half for server A and half for B:

```
AVSTAR-A#  offline
AVSTAR-A#  configure
AVSTAR-A#  online
```

Wait for the system being configured messages to appear on both sides of the console before moving on to the next step.

**Step 13: Start up the revived server.**

When you run `startup` on the revived server, the devices and utility programs (such as action servers or txnet links) on that server are started and it is placed online:

```
AVSTAR-B# startup
```

**Step 14: Put the master computer back online.**

The system is now running normally with A processes and devices running on server A, and B processes and devices running on server B. The `diskcopy` command will continue to run in the background, working to mirror the entire database. The `diskcopy` process will send progress messages to the console screen and eventually report that the `diskcopy` is complete. You can check whether it has completed by running the `status` command. The disk status will be UNKNOWN until `diskcopy` completes; at that time it changes to OK.

**Step 15: (If you used the alternate method 2 in step 10 and typed logout all and stop all)**

If you decided to log out all users and stop all devices in step 10, you should restart all of the survivor's devices by typing `restart all` on the master computer.

The system is now running normally (dual) with all sessions and devices in their normal places. The `diskcopy` may still be running in the background, copying the database back to the failed server.

If server B was selected as the master database, it is now the master computer. Since either server can run as master, nothing further needs to be done. If you wish to make server A the master computer again, wait until the `diskcopy` completes and then perform a normal system shutdown, reboot and startup, as described in Chapter 3.



**Wait for the `diskcopy` to complete before rebooting the system to make server A the master computer again. Database corruption may result if the system is taken down before the `diskcopy` completes.**



If the system loses power or must be rebooted; the master computer that contains the full database must be brought up in a single server configuration, and the disconnect recovery procedure restarted again.

## Worksheet

The following Disconnect Recovery table lists:

- Commands involved with recovering from a disconnect
- Whether the command is run on the failed server, survivor, or both servers
- Offers a column where you can fill in the name (either A or B) of the server that fits that role (either survivor or failed)

You may want to photocopy this table and write in names (letters) of your survivor and failed servers in the boxes for each step.

Before proceeding, review and understand the narrative steps in the previous section "Procedures" on page 21-8.

The table's procedure shows Method 2 (logout all, stop all) of splitting the workload to get the devices, utility programs, and sessions back to their regular positions.

**Table 21-1 Disconnect Recovery Worksheet**

Computer	A/B	Commands to Run:
survivor	<input type="checkbox"/>	<i>become superuser</i> <b>offline</b> <b>configure</b> <b>online</b> <i>reboot failed server's network devices</i> <i>restart failed server's network devices</i>

Computer	A/B	Commands to Run:
failed	<input type="checkbox"/>	<pre>sync init 6 login as system operator become superuser diskclear -</pre>
both	AB	<pre>reconnect &lt;failed&gt; master=&lt;survivor&gt; net=ab type either: reconnect a master=b net=ab -OR- reconnect b master=a net=ab</pre>
failed	<input type="checkbox"/>	<pre>diskcopy -s</pre>
survivor	<input type="checkbox"/>	<pre>offline logout all stop all configure</pre>
both	AB	<pre>wait for the "system being configured" messages on both</pre>
survivor	<input type="checkbox"/>	<pre>restart all online</pre>
failed	<input type="checkbox"/>	<pre>startup be patient for the server to come up</pre>

# CHAPTER 22

## *Troubleshooting*

This chapter provides information to help you recover from various kinds of system failures. The major sections are:

- Avstar Workstation Problems
- Busy Stories
- Wire Problems
- Printer Problems
- Locked Blocks
- How to Check Process Status
- Power Failure
- Hard Drive Failure
- Network Failure

## Avstar Workstation Problems

This section explains common problems users may have during daily operation. Possible solutions appear after each problem.



*A user who installs the client software, Avstar Workstation (ASWS), on a computer running Windows NT operating system must have permission to overwrite system files necessary to run the product (as in MFC42.DLL). Log in as administrator and set privileges on user accounts to enable access and modification rights on the system32 directory.*

### A User Cannot Log in

If a user is unable to log in to Avstar NRCS and gets the `Invalid User Name/Password` error message, ensure the user:

- Entered correct user name and password
- Specified proper server name
- Spelled name and password correctly

Use the `list u` console command to verify a user with that name exists on the system.

If the password is a problem, assign the user a new password. Then, force a password change to maintain security. See Chapter 18, “Database Security,” for more information.

## A User Cannot Establish a Session

If the user is properly entering user name, password and server name, but the login attempt “hangs” and then begins to cycle through alternate server names, wait until the login attempt times out and follow these steps:

1. Try to ping the server to check network connection to the server, by doing the following.
  - a. Click Start.
  - b. Select Programs.
  - c. Select MSDOS Prompt (or Command Prompt on Windows NT) to open a DOS window.
  - d. Enter the following command, substituting the name of your server:

```
C:\>ping avstar-a
```

A message similar to the following should appear:

```
Pinging avstar-a [10.1.1.38.30] with 32 bytes of
data:

Reply from 10.1.1.38.30: bytes=32 time<10ms TTL=255
Reply from 10.1.1.38.30: bytes=32 time<10ms TTL=255
Reply from 10.1.1.38.30: bytes=32 time<10ms TTL=255
Reply from 10.1.1.38.30: bytes=32 time<10ms TTL=255

C:\>;
```

If you do not get replies, then the server is either down or a networking problem exists. If you get a message similar to:

```
Bad IP address avstar-a
```

-OR-

```
Bad host avstar-a
```

Avstar Workstation is unable to look up the name to obtain the IP address of the server. It is unable to resolve the host name. This indicates the name entered in the login dialog box does not exist in the local hosts file on the PC or on the Domain Name Server (DNS) if your network is configured to use a DNS.

2. If the workstation can successfully ping the server, the next step is to run the `workdebug stat` command on each server at the Avstar console.

The `workserver` process runs on Avstar Servers that manage the connection between PCs and servers. The `workdebug stat` command queries the `workserver` to obtain a list of how many sessions are configured and how many are currently logged in:

```
AVSTAR-A# workdebug stat
workserver:configured DOS general count:A 20 B 0 C 0 D 0
workserver:configured DOS general count:A 20 B 0 C 0 D 0
workserver:running DOS general count:  A 0 B 0 C 0 D 0
workserver:configured gui general count:A 19 B 0 C 0 D 0
workserver:running gui general count:  A 0 B 0 C 0 D 0
(etc)
```

If `workdebug stat` does not return a response at the Avstar console, this may indicate the `workserver` process has either died or entered a non-responsive state. The `workserver` process may need to be killed and restarted. Contact iNews Customer Support for further instructions.

3. If you get a response from `workdebug stat`, you can put `workserver` into a diagnostic mode where it will display messages on the screen when PC clients attempt to establish a connection. To do this, type:

```
AVSTAR-A: workdebug debug
```

The message returned when the workstation attempts to log in may give you a clue as to the problem. If no messages appear when the workstation tries to log in, then the workstation is not reaching the server. Use the `workdebug` command again to turn

off debugging mode and stop the diagnostic messaging, by typing:

```
AVSTAR-A: workdebug silent
```

## A User Cannot Access an Item

### Group Permissions

If a user cannot read or write stories in part of the database, he or she may not have permission to do so.

To assign permission to a user, do the following:

1. Use the `list g <username>` command to determine which groups the user belongs to.
2. Compare assigned user groups to groups assigned to the directory or queue. To do this, use the `list d-g <name of queue>` command.
3. If necessary, add the user to the appropriate group story or stories in `SYSTEM.GROUPS`. See “Adding Members to an Existing Group” on page 6-15 for more information.

If a user can create new stories in a queue, but not edit existing stories, it may be because existing stories were created when the queue had another write group assigned to it. Previous stories would have been created with previous security restrictions. To change the group permissions on existing stories, use the `gtraits` command in the following format:

```
gtraits c <name of queue>
```

A user may also be unable to access a queue or story in the database if another user is ordering that queue or editing the story. The queue or story becomes “free” when the other user finishes the operation, but until then, it is considered to be “busy.”

## Busy Stories

Only one user can edit a story at a time. No one else can edit that story until the first user is done with it. If a second user tries to edit a story that another user is working on, the second user will get a message that the story is busy. When a user opens a story for editing, the system puts an edit lock on the story and removes it when the user saves the story and gets out of it. Edit locks prevent multiple users from making changes to stories.

Similarly, when a user goes into order mode, the system puts an order lock on the queue. No other user can change the order of stories around in the queue until the first user exits order mode.

If a user is editing a story and the system crashes, or the user's PC locks and needs to be rebooted, the edit lock placed on the story remains attached to the story. When the system comes back up or after the user logs back in, he or she will be unable to edit the story. The edit lock will prevent anyone from making changes to the story and users trying to open the story will get a `story busy` message.

The story must be unbusy before any user can get back into it.

This is also true of order lock. If a user's PC crashes while he or she is in order mode, the order lock remains behind. Both edit locks and order locks are removed with the `unbusy` command. The syntax is:

```
unbusy <queuename>
```

You must know the exact queue name to unbusy it, such as `SHOW . 6PM . RUNDOWN`.

When you unbusy a queue and there is an order lock on it, you are first asked whether you want to remove the order lock. Then you are asked whether you want to unbusy each busy story in the queue.

You can respond with `yes` to remove the edit lock or order lock, `no` to skip that story, or `quit` to exit.





Care should be taken when removing edit locks. Do not unbusy stories that users are still working in. If you do, when they try to save the story it will be saved to the Dead queue.

You may see messages on the Avstar console about edit-locked (busy) stories in the Dead queue. A large number of edit-locked stories at the bottom of the Dead queue can cause problems, and they should be unbusied.

The Dead queue is usually large and an unbusy on the entire queue could take a long time to execute. If you need to unbusy edit-locked stories in the Dead queue, do the following:

1. Become a superuser. See “Becoming a Console Superuser” on page 3-2 for more information.
2. Un-invert the Dead queue, by typing:
 

```
AVSTAR-A# dbtraits dead -i
```
3. Use the `unbusy` command to remove the edit locks.
 

```
AVSTAR-A# unbusy dead
```
4. The system will prompt you for confirmation. After answering “yes” to unbusy the stories, watch for when the system is no longer finding busy stories. You can then break out of the unbusy process with **CTRL-\** (Control-backslash).
5. Re-invert the queue to put it back to its original state.
 

```
AVSTAR-A# dbtraits dead +i
```



*On SGI systems, the prompt sometimes acts inconsistently after breaking out of a process. If so, press **CTRL-D** a few times to restore normal operation.*

## Wire Problems

Wire ports are either on or off. They either receive data and stories or they do not. The wire port cannot decide to let some stories in but not other stories. If stories are coming in to your AP wire port but it is

missing some—weather stories for instance—this is most likely a problem with what AP equipment is feeding because the PCU/CCU port is not selective about which stories it decides to keep and which to ignore.

If a wire stops flowing into the system entirely, the first step is to figure out which wire port is not working. On its line in the `/site/config` file, each wire port is assigned a name, such as AP, RT, or NC. This name is in the fifth column of the wire's configuration line. For instance, the following configuration file entry shows a wire port with the name AX:

```
wire    23    9600-8n    anpa7    AX - ;Associated Press
```

You can check `WIRES.ALL` and look at the coding on the stories to determine which port is not flowing, and then restart that wire port. For the example above, stories that come in port 23 will have coding that begins with AX in the writer field:

Title	Writer	Created
AP-IN--FWA-SevereThunder	AXv2in-gob	07/06/00 14:54:53

If you notice AX-named wires had not come in for a period of time, wire port 23 must be restarted and then the Wires queues monitored to see whether the wire is working again.

Alternatively, you can restart the wire port with the minus four (-4) option:

```
AVSTAR-A: restart -4 23
```

When restarted in this diagnostic mode, a message is printed to the console screen each time a wire story arrives and is placed in a Wires queue. Restart it normally once you have confirmed wire flow.

If it is still not flowing, get instruction from your wire service provider on how to reset their equipment. Resetting the wire's modem should trigger a wire reset message which should arrive in either the `WIRES.ALL` or `WIRES.UNKNOWN` queue. If the reset message reaches

the database, it is an indication the wire port is working, and wire stories should then start to flow.

If you do not get wire flow, try stopping, resetting (power cycling), and then restarting the entire PCU/CCU.

You can also do a wire dump, which redirects any characters coming in the PCU/CCU port and displays them on the console screen. If you are getting no wires and no wire dump, it may be possible the wire modem is feeding the signal with the wrong polarity.

The only way to determine polarity is to hook up a mini-tracker and examine the pin 3 light as data is received. Pin 3 should flicker green as data flows across the connection. The PCU/CCU port will not recognize red data flickering. Red data indicates incorrect polarity coming from the wire equipment and must be rectified by your wire service provider.



*Red data will appear on terminals and printers hooked to the wire output, but it will not be recognized as data by the PCU/CCU port.*

Each wire port must have a corresponding wire profile in `/site/wires/<port #>` if you need to move a wire to another PCU/CCU port. For instance, if you move a wire from port 17 to port 38, you must also copy its profile to the new port number with the Copy (`cp`) command:

```
cp /site/wires/17 /site/wires/38
```

## System Printer Problems

When a story is printed to a system printer, the print job goes to the appropriate `SYSTEM.PRINTERS` queue.

The printer port communicates with the printer using `xon/xoff` flow control. The printer port starts sending a print job out the port and waits to hear back from the printer. If the printer's buffer is getting full,

the printer sends an xoff signal to the port, telling it to stop sending data. When the printer is ready to accept more of the print job it sends an xon signal, telling the port to resume data flow.

If the CCU port does not receive an xoff signal, it assumes the printer is processing the print job and pipes it out the port. It stops only if the printer tells it to stop. If the printer is broken or turned off, it will not send the xoff to the CCU, so the CCU will send the print job out the port.

If the printer is off-line, it will transmit an xoff signal back to the CCU port when the port tries to send a print job, and print jobs will back up in the `SYSTEM.PRINTERS` queue. The system will try to print the job for up to three minutes. After that time a message will be returned to the user stating the printer is off-line and needs attention. A similar message appears on the console.

Occasionally, a user will print an entire queue, such as `WIRES.ALL`. This is called a *runaway* print job and it must be deleted from the `SYSTEM.PRINTERS` queue.

To eliminate a runaway print job, do the following:

1. Power off the printer or put it offline so it stops wasting paper.
2. Kill the print jobs in the `SYSTEM.PRINTERS` queue.
3. Restart the printer port.
4. Turn the printer back on.



*You must restart the printer port because the print job has already been downloaded to the PCU/CCU for printing. If you kill the print jobs and then turn on the printer, it will resume printing the job.*

Each printer port must have a corresponding printer profile in `/site/printers/<port #>`. For instance, if you move a printer from port 26 to port 41, you must also copy its profile to the new port number:

```
AVSTAR-A: cp /site/printers/26 /site/printers/41
```

## Locked Blocks

During normal operations the system constantly locks and unlocks blocks on the hard drive as data is accessed. Occasionally, something may go wrong and a locked block will not unlock. A locked block is left behind.

If you have a persistently locked block, any other process that tries to access the locked block will also lock up waiting for the block to become available. The usual scenario for this problem is that PCs and terminals start locking in the newsroom when users get to a story in a rundown that is in a locked block. Users then move to another PC or terminal, and lock that one up too when they access the locked block story.

If the original locked block is unlocked, then all frozen PCs and terminals “unstick.”

To check for a locked block, type the `dblock` command on all servers. Type `dblock` several times in succession and note the block numbers reported. If block numbers are changing, then the process is not hung up on a locked block. If the block number and process ID number remain the same through successive `dblock` commands, then it may be a locked block situation.

If you have a persistently locked block, contact iNews Customer Support.

## How to Check Process Status(`ps` Command)

Occasionally, the support staff may ask you to run a few UNIX commands to troubleshoot problems. One of the most common would be a request to check on the status of a process using the `ps` command.

```
ps -ef
```

The `ps -ef` command returns a list of all the processes currently running on the server. On large systems there can be hundreds or even thousands of processes.

The process list gives useful information about each process including the process owner, when it was started, on which terminal port it started, how much processor time has been expended on the process, and so forth.

Because it produces such a large list, on SCO and SGI systems you can go into the shell (by typing `sh`) and use the pipe symbol (`|`) to place the output of the process list into the `more` command. This breaks the `ps` list into screen-sized pieces rather than scrolling it up off the top of the screen:

```
AVSTAR-A# sh
ps -ef | more
```

When you are done looking at the current screen, press the spacebar or Enter key for the next screen.

You can also use piping with the `fgrep` command to search the process list for a particular process and check whether it is running. This example produces a full process list and then filters it for the lines that contain the word, `workserver`:

```
AVSTAR-A# sh
ps -ef | fgrep workserver
```

When you use `fgrep` on the process list for a specific word, two processes are usually returned in the display. One line is the process you are looking for, and one line is the `fgrep` process that looked for that word.

If you know a particular process ID number, you can check to see if that process is running or has completed:

```
ps -p <pid #>
```

If the process is still running, you will get a one-line process list for that number:

```
PID TTY      TIME CMD
516 tablet  0:01 workserver
```

If it is not running, the command will simply return a header line like this:

```
PID TTY      TIME CMD
```

You may also use `ps -fp` for a fuller listing.

## Power Failure

If you experience a power failure, the servers will reboot. After they work their way back to the login prompt and you have logged in as superuser, the servers will not be named, they will be at the question mark-colon prompt.

```
?:
```

If both servers went down at the same time, the databases will still be in mirror and they can be connected normally and started up:

See “Selecting Servers” on page 2-5 for more information.

1. On *all* servers simultaneously, type:
 

```
connect #
```
2. To remove edit locks, on the master computer only, type:
 

```
dbclean -x .
```
3. On *all* servers simultaneously, type:
 

```
startup
```

If they are plugged into different UPSs and they ran through the UPS battery and then lost power and rebooted, you will not know if they went down concurrently. One UPS may have run longer than another. If so, more stories may have flowed in on a wire or been saved by a

user while the one server was still up and the other was down. The database will not be in mirror and you will need to go through the recovery process as noted in Chapter 21.

## Hard Drive Failure

If you lose a hard drive in a server and the server will not boot, you can restore the software to a new hard drive. Call iNews Customer Support for assistance restoring the software.

You will need the following:

SCO systems     `softdump` tape for that server

emergency boot floppy

emergency root floppy

SGI systems     `softdump` tape for that server

IRIX 6.5 Release CD

If you do not have the required tapes, the operating system software will need to be reloaded from scratch. This procedure could require the servers' return to iNews for reload or a site visit.

## Network Failure

Avstar NRCS is a networked client-server application. A well-running Ethernet network is essential for proper communication between the devices. A network failure can disable the entire system.



The simplest way to test network connectivity is to try to ping other computers on the network. When you ping a server, your computer sends a “pulse” across the network. The pulse then bounces off the target server and returns to the sending computer in the form of ICMP replies. If the network is down, you will not get replies from the server (or computer) you are trying to ping.

```
AVSTAR-A: ping 152.165.17.110
```

-OR-

```
AVSTAR-A: ping avstar-a
```

In the case of an Avstar Workstation that is having problems connecting, the first step is to try pinging the server from the workstation to make sure they can see each other on the network.

If computers are unable to ping each other, check for a loose or disconnected network cable or a hub that may have lost power.

## netstat -i Command

One diagnostic command you can run on the server to quantify network errors is the `netstat -i` command. The `netstat -i` will show:

- How many packets have been transmitted
- How many input errors (`Ierrs`) have been detected
- How many output errors (`Oerrs`) have occurred

## Output Errors (Oerrs)

The primary barometer of network health is the Output Errors (`Oerrs`) column. The computer will try to transmit a packet 15 times before chalking the attempt up as an `Oerr`. It will then try to transmit again 15 times and may wind up incrementing the `Oerr` count. Clean networks will show zero `Oerrs`, or no more than a few.

Of more concern than the raw number of `Oerrs` is how quickly they are increasing. If you are picking up an `Oerr` every minute, this would be indicative of network problems and a disconnect may be imminent.

Periodically run a `netstat -i` so you can get a baseline feel for how many `Oerrs` your system produces each week or month.

### **Input Errors (`Ierrs`)**

Input Errors (`Ierrs`) are fragments of packets or unrecognizable packets. Systems co-existing on a Novell Local Area Network (LAN) generally show many `Ierrs`, although they do not seem to cause problems.

# SECTION IV

## *System Reference*

This section contains the following appendices:

- Appendix A, *Command Reference*
- Appendix B, *System Files*
- Appendix C, *Standard Dictionaries*
- Appendix D, *PCU References*
- Appendix E, *Character Mapping Codes for Wires*
- Appendix F, *Environment Variables*
- Appendix G, *Managing Traits at the Console*
- Glossary

---

# APPENDIX A

## *Command References*

Although a few of your system's commands are UNIX commands, most are special commands provided by Avstar.

SCO Systems

On DEC/MIPS systems, many UNIX commands have been removed to provide as much disk space as possible for your database.



**Some available commands are meant to be used only by iNews technicians or under the supervision of iNews personnel. These commands may cause damage if used improperly. They are listed in Table A-1 and Table A-2 on page A-3.**

This appendix lists and explains commands available on your system. Examples are also included for commands you are most likely to need. It contains the following sections:

- Programs Invoked by Avstar
- Commands Used by iNews personnel Only
- UNIX Commands Used in Avstar
- Console Server Commands
- Job List Commands
- Dialog Commands

## Programs Invoked by Avstar

The following programs are invoked and used by your Avstar system. Do not use them as commands.

**Table A-1 Programs Used by Avstar**

action	bioserver	boot
brand	connect.sh	console
copyright	diddle	disconnect
distribution	gnews	keyword
license	mailserver	netterm
news	newsmail	nxserver
parallel	rxnet	seek
server	snews	softcheck
start	telex	txnet
workserver		

## Commands Used by iNews personnel Only

The commands listed in Table A-2 are used by iNews personnel only.

**Table A-2**      **Commands Used Only by iNews personnel**

attach	balloc	bdump
bex	binhex	biodebug
bioq	biosleep	biostat
bprof	catcheck	ccputwire
ccuq	ccuspeed	ccutime
dbcopy	dbfind	dbprint
dbsize	detach	download
finit	ifis	ifmaster
ifsu	iftapeis	keycheck
kwdcheck	load	makens
makeform	mpdebug	mpdebug
nsupgrade	nsupgrade	nsupgrade
qcheck	qcheck	qcheck
rcat	sosh	traverse
uprefcheck	userclean	wdump
windebug	workdebug	wxlate
xi		

## UNIX Commands Used in Avstar

The following UNIX commands are available in Avstar. For further information, see the reference material that came with your UNIX system. To obtain command syntax and other usage information, type the `man` command along with the command name. For instance, to get information about the `grep` command, type:

```
man grep
```

**Table A-3** UNIX Commands Available in Avstar

cat	cp	date
df	grep	halt or haltsys
init 0	kill	mv
newfs	passwd	ps ax
ps -ef	pwd	rm
sh	sync	telnet

## Console Server Commands

You must enter commands in lowercase. Your system does not recognize commands entered in uppercase.

```
broadcast broadcast <message>
```

Sends a message to everyone logged in. For instance, to send a message, select *one* server and type:

```
AVSTAR-A: broadcast System going down at 12:00
```

```
ccputkey ccputkey <terminal #> <string1> ...
```

Executes workstation commands as if they were entered at a CCU (non-network) workstation. Follow the `ccputkey` with the device number of the workstation you want to use and the command you want to execute.



Your command must include the Command and Return keystrokes, represented by `\C` and `\r` respectively.

For instance, to script a story on workstation 51, type:

```
ccputkey 51 \Cscript\r
```



**Do not use this command on a workstation where someone is working, because you may damage their work.**

`configure`

```
configure -[n|s] [<config file> [<system>
<computer>]]
```

Incorporates changes to your configuration file into your system's operation, and checks the configuration file for any errors.

For instance, suppose you made changes to PCUs 10 and 20, which are connected to server A in an AB system. To test these changes, become a superuser and type:

```
AVSTAR-A# configure /site/config ab a
```

If a service has been added to a database story in `SYSTEM.SERVICE`, use `configure -s` so the service can be recognized.

If an Ethernet or Internet address has been added to a database story in the `SYSTEM.AVSTAR.DOS` or `SYSTEM.AVSTAR.WINDOWS` directories, use `configure n` to validate the address and allow it to be recognized by the system.

In both cases, you must first take the system offline, enter the `configure` command, then put the system back online.

The `configure` command changes to support the device you want to add. See the following series of configuration lines to enter a `configure` command for a specific device.

```

host      <system> <computer>
          serial    <port 1> ...
          net       <network name> <net ccu 1> ...
          servers   <server 1> ...
ccu       <devid> <speed> <type> <port 1> ... <port 8>
ccu       <devid> <ccuname> <pcuname> at <port 1> ...
<port 8>
printer   <devid> <speed> <printer number>
server    <devid> <name> <notify> <devname>
special   <devid> <speed> <name> <mailbox> <devname>
terminal  <devid> <speed> <printer number> <name>
<device name>
dialup    <devid> <speed> <0> <name> <device name>
unused    <devid>
wire      <devid> <speed> <program name> <source name>
<'join', 'raw', or '-'>
line      <devid> <speed> <name> <device type>
resource  <devid> <name> <resource i/o device>
driver    <devid> <speed> <program name> <device name>
mcspc     <devid> <pcname> <program name> <device name>
avstar    <devid> <ethernet address> <printer number> <name>
<device name> <dos>
nswin    <devid> <ethadd> <print> <name> ... <windows>

```

```
connect    connect <name> [<option>=<value>] ...
```

The `connect` command names each server in the system, and tells each how many other servers there are in the system and how to communicate with them. For instance:

```
AVSTAR-A: connect a net=ab
```

**Table A-4 Options for the Connect Command**

<code>airmargin=&lt;number of columns&gt;</code>	<code>airshift=&lt;yes   no&gt;</code>
<code>auto_upgrade=&lt;yes   no&gt;</code>	<code>broadcast=&lt;mm:ss&gt;</code>

**Table A-4 Options for the Connect Command**

dialect=<number>	disk=<status>
highwater=<hundreds_of_blocks>	id=<system name>
language=<number>	lastlogin=<yes   no>
localtimeout=<mm:ss>	logintimeout=<mm:ssc>
lowwater=<hundreds_of_blocks>	master=<name>
maxhits=<number>	min_passwd_length=<number>
msgserver=<silent   verbose>	name=<a   b   c   d>
netlogintimeout=<mm:ss>	nsload=<number>
outtime=<mm:ss> or <hh:mm:ss>	pausetimeout=<mm:ss>
purgelimit=<number of hours>	readrate=<number of words per minute>
remotetimeout=<mm:ss>	security=<and   or>
scriptlthmax=<number of columns>	scriptrhmax=<number of columns>
single=<name> or net=<name,name[,name,name]>	textmax=<number of columns>
timechar=<character>	timer=<silent   verbose>
timing_form=<form name>	

dbclean

dbclean [-x] <directory name>

When starting up after a power failure, use `dbclean` to remove any edit or order locks in the database. Run the command before startup. Log everyone off the system by typing `logout all` before issuing this command.

The most common usage of `dbclean` scans all queues except those marked with the skip flag. To use this command, after logging out all the users, type:

```
AVSTAR-A: logout all
AVSTAR-A: dbclean -x .
```

dbclose

dbclose

Closes the database.



**If you use this command while users are active, changes to stories will be lost.**

dbdev

dbdev

Reports the disks in use and the number of blocks allocated for the database on each disk. To find out the size of your database, type:

**dbdev**

A message similar to the following appears:

```
/dev/rp5  0          60000
/dev/rp6  60001     120123
/dev/rp7  120124    999999
```

The first number of the report (60,000 in the example above) is the size of the database.

dbdump

dbdump &lt;keys&gt; [&lt;option&gt;] ...

Dumps individual stories or the entire database to tape.

This command can be interrupted. The program will continue dumping to reach an appropriate quitting point.

**Table A-5 Valid Keys for the dbdump Command**

Valid Keys	Description
a	Append to current tape
c	Create new tape
d	Dump the Avstar Directory
i	Dump indexed files (such as, user index)
t	Show titles of dumped stories
v	Verbose output
x	Ask before dumping indexed file

**Table A-6 Valid Options for the dbdump Command**

Valid Options	Description
-n <namelist>	Only dump listed directories
-m <minutes>	Dump files modified in last x minutes
-N <computer name>	Network dump
-a <device>	Use alternate device for dump
-f <device>	Dump to disk device (/dev/diskpack)

To dump everything except those directories marked with a skip flag, type:

```
dbdump c
```

To dump a queue to a new tape, add `-n` and the queue name to the command. For instance, to dump the queue `SCRIPTS.JUNE.01` to a new tape, type:

```
dbdump c -n scripts.june.01
```

By replacing the `c` with an `a`, you can add a queue to a tape without erasing information already on the tape. For instance, to append `SCRIPTS.JUNE.10` to a tape, type:

```
dbdump a -n scripts.june.10
```

`dbfree`

`dbfree`

Reports the size of the database and the size of the free list—that is, the amount of free blocks available. To display this information, type:

```
dbfree
```

A message similar to the following appears:

```
data base size (150000) free blocks (12000)
```

To display the amount of free space in your software area, type:

```
df
```

On SCO systems, type: `df -v`

`dblines`

```
dblines [c | n | q | s | v] <pathname>
```

Checks the database for story errors.

**Table A-7 Valid Keys for the `dblines` Command**

Valid Keys	Description
<code>c</code>	Complete check
<code>n</code>	Do <i>not</i> fix errors
<code>q</code>	List queue names
<code>s</code>	Skip queues that are skipped by <code>dbdump</code>
<code>v</code>	Verbose output

Use this command weekly as part of normal database maintenance. Start `dblines` before you go home and run it in the background. To run `dblines` in the background, precede `dblines` with a `print`

command. Including a period with the command checks the entire database.

For instance, to run `dblines` in the background and send any error messages to printer 2, type:

```
print 2 dblines
```

If `dblines` finds any errors, call iNews Customer Support for assistance in removing them.

`dboriginal`

```
dboriginal <pathname>
```

Removes all old versions of stories in a queue to the freelist, so use it only on queues where you do not need to retain these old versions. For instance, to remove the old story versions in `ARCHIVE.MARCH`, type:

```
dboriginal archive.march
```

Use the `dboriginal` command to reclaim space when the system is low on space.

`dbpurge`  
(Superuser  
only)

```
dbpurge <path> [- h | l | v | f] [<interval>]
```

`dbpurge` purges the database to regain space.

**Table A-8 Valid Keys for the dbpurge Command**

Valid Keys	Description
-	Purges all queues with the default interval
-v	Purge all queues listed in the command in verbose mode
-h	Purge held entries (must be superuser to use this option)
-l	Purge locked entries (must be superuser to use this option)
-f	Purge future dated entries (must be superuser to use this option)
	Interval is expressed in hhh or dd . hh

The system automatically purges the directories and queues each hour to move old material from high-turnover queues (like the Wires queue) into the Dead queue. Use this command only in an emergency

when you need to regain some space. Stories can receive a date in the future if your system date is inaccurate. If you have future-dated stories because the system date was inaccurately set, remove them using the `-f` option or wait until the date expires. You must be a console superuser to run `dbpurge` on write-protected queues. For instance, to purge all stories in the Wires queue older than five hours, type:

```
dbpurge wires 5
```

Typing `dbpurge -` purges all queues in the database. `dbpurge -v` does the same, and prints a message on the console for each queue purged.

You can also use `dbpurge` to remove held and locked stories from the database. To remove all locked stories from all queues in the People directory, type:

```
dbpurge people -l
```

To remove all held stories from a queue, use `dbpurge`, but substitute `-h` for `-l`.



*The `dbpurge` command can be run only on the master computer.*

`dbrestore`

```
dbrestore <key> [option] ...
```

Converts `dbdump` input to the current database revision.

Existing form numbers stored in directory records and in stories will be converted to form names consisting of three digits. See the `dsform`, `dqform`, and `sform` data elements.

If your database is damaged, you can restore it from your backup tape by typing:

```
dbrestore div
```

When you use `dbrestore`, restoring large numbers of stories can cause a temporary out-of-space condition.

Press Delete to stop a `dbrestore` in progress.



**Table A-9 dbrestore Command Key Options**

-a <device>	Restore from alternate device
-c <filename>	Character conversion profile filename
-C <filename>	Character map filename
-d	<date>[-<date>]
-f <device>	Restore from disk device
-k	<keyword> (You can specify multiple keywords.)
-m <value>	Maximum number of stories to restore
-M	Preserve modification times
-n	<directory>[=<new name>]
-N	Read from network socket for dump data
-p <queue>	Only with key letter s, will create queue
-s <platform>	SGI, MIP, SCO

**Key May Contain:**

s	Restore stories only
d	Restore stories with their original directories
i	Restore ISAM files
v	Verbose output; list directory names
vv	List directory names and <i>each</i> title
x	Ask before restoring each ISAM file
f	Print facts about blocks and times
t	Print table of contents; do not restore



*You cannot use both the s and d options in the same command. Select one or the other; not both. Additionally, you must be a superuser to list ISAM files.*

The `dbrestore` character mapping feature allows a single character to be mapped to one or two characters. The syntax of the character mapping file supplied with the `-C` command option to `dbrestore` is extended for the single character to two-character mapping.

See “dbrestore Conversion Map” on page E-15 for more information.

Map definitions now come in three forms:

65 -> 97	(a single to single map)
65 - 90 -> 97 - 122	(a range of single to single maps)
247 => 225 194	(a single to two-character map)



*Notice that the difference of mapping to a single versus a double is an equal sign (=) instead of a dash (-).*

`dbserver`

`dbserver <high water>`

Reclaims space from the Dead queue and places it on the free list. Use `dbserver` to build up the free list prior to periods of peak use. The `dbserver` command reclaims space more actively than the automatic `dbpurge` by looking for out-of-date stories to remove.

When you use `dbserver`, specify how many free-list blocks to reclaim from the Dead queue. Each free-list block is equivalent to 100 database blocks. For instance, to build the free list up by 5,000 database blocks, type:

**`dbserver 50`**

You can also use `dbserver` to place all space in the Dead queue on the free list. To do so, specify that you want to rebuild the free list to an unreasonably large size. For instance, type:

**`dbserver 10000`**

The `dbserver` command is invoked when the system is booted and runs in the background if the database does not satisfy the high-water and low-water restraints.

`dbsort`

`dbsort [-v] <queue name>`

Sorts stories in any queue by the sort field for that queue. For instance, if `RUNDOWN.AM` has the page number field set as its sort field, type the following to sort the queue by page number:

**`dbsort rundown.am`**

If no sort field has been set for the queue, its stories are sorted by the value of the title field. If a sorted queue is ordered, the sorting is disabled. Using `dbsort` starts the sorting again. Only a superuser can sort queues with nonzero write groups.

`-v` makes stories in queue adopt a sort field, but does not actually provide a sorting function.

Use the `-v` option to verify the sort field. The system checks that the quick index field in the database has the same data as the sort field in the story. This option provides no sorting function, but it updates the sort field so that your next sort is based on current information.

`dbtraits`

```
dbtraits <pathname> [only] [<option> <value>]
[+|- mode] ...:
```

**Table A-10 dbtraits Command Options**

<code>abstractlines</code> or <code>al</code>	<code>queueform</code> or <code>qform</code>
<code>abstractprinter</code> or <code>ap</code>	<code>save</code>
<code>abstractstyle</code> or <code>as</code>	<code>sortfield</code> or <code>sf</code>
<code>changeform</code> or <code>cform</code>	<code>storyform</code> or <code>sform</code>
<code>mailbox</code> or <code>mb</code>	<code>stripform</code>
<code>purgeinterval</code> <days.hours>	<code>writgroup</code> or <code>wg</code>
<code>readgroup</code> or <code>rg</code>	<code>writgroup</code> or <code>wg</code>

Sets and modifies database traits.

To assign form 5pm to the Rundowns directory, type:

```
dbtraits rundowns.5pm form 5pm
```

**Table A-11 dbtraits Command Modes**

g	General
i	Inverted
p	Printable
q	Queue operations allowed
r	Read access
refresh	Available only on systems that allow queue refresh
s	Sequential
so	Sorted
u	Update
x	Skip

dbvisit

dbvisit `[-d | v] [-r | m name] [-s] [block# ...`**Table A-12 dbvisit Command Options**

r	Read only; do not rebuild free list
s	Operate in slow mode to eliminate cache usage
m	Machine name to disconnect (for online use)
-d	Print dot for every queue visited
-v	Verbose output; print name of each queue

Scans the database for errors, then rebuilds the free list and fixes bad story-link counts. Use `dbvisit` once a month as a part of your regular maintenance.

There are two ways to run `dbvisit`:

- Without the `-m` option: The system must be offline and shut down. No one can log in until `dbvisit` is complete.
- With the `-m` option: Specify a machine that will be running `dbvisit`. The system must be offline and shutdown only long enough to enter the `dbvisit` command. Remaining machines can

be brought up for users to log into. Once the `dbvisit` procedure is complete, the machine running `dbvisit` can be reconnected following the normal procedure.



**Before you run `dbvisit`, ensure the system is offline, all PCUs/CCUs are stopped, and the system is shut down. If `dbvisit` reports any errors, do not rebuild the free list; call iNews Customer Support for assistance.**

```
dial /exc/dial -p<port> [-i<devid> -o<options> -
b|c|d|e|i|q|S|t|x| -m<modem> -n<phone#>]
```

This command makes a connection between a workstation and a modem, passing data in both directions. Use it only in the service table, not from the console.

**Table A-13 dial Command Options**

-b	Binary data
-c	Check evidence of modem
-d	Direct (no modem)
-e	Echo
-i	Device ID for a direct-line device
-m	Modem type
-n	Phone number
-o	Communications options
-p	Port number
-q	Allow user to quit with Control <space>
-S	Special device
-t	Touch tone
-b	Binary data

diskclear  
(Superuser  
only)

diskclear [-|u]

Marks each block of a server's database as invalid so that you can copy a new database to the disk. To clear the disk, select a server and type:

**diskclear -**

Do this prior to connecting a replacement server to a running system.



**Using this command erases the server's entire database.**

To reverse the effects of diskclear, type:

**diskclear u**

**Table A-14 diskclear Command Arguments**

Arguments	Description
-	Clear database
++	Read and write to disk (cannot be repaired by running diskclear -u)
--	Read database blocks
<number>	The number of blocks to process at one time
u	Unclear database

This command works after you have cleared the disk and before you have reconnected the server and begun copying a new database.

diskcopy

diskcopy -[n] n=number of simultaneous copies per  
disk diskcopy <start block> [<end block>] diskcopy  
-s

Copies database from master computer (usually server A) to a replacement computer. Enter it on the replacement server.

Select the **-s** parameter to minimize impact on system performance.

```
doc          doc -g <queue> [<title>] <file> (gets file from
            database)
doc          doc -p <queue> [<file>] (puts file into database)

title must be one word or enclosed in quotes.
```

Run this command from the shell, not from the Avstar prompt. If `<file>` is not specified, the story containing the specified title in the specified queue will be displayed on the console.



**This command transfers files between the database and software areas of a disk. Use it only if instructed to do so by iNews personnel.**

```
ed          ed <file pathname>
```

This command initiates the UNIX line editor used to edit UNIX text files. Since each server has its own copy of each site file, always select *all* servers before editing a site file. Procedures for using this line editor is covered in detail in Chapter 10.

```
extract
(Superuser
only)
```

```
extract [-a <device>] [<config file>]
```

**Table A-15 extract Command Options**

<code>extract [-a &lt;device&gt;] + [&lt;config file&gt;]</code>	Echo list of files that need to be extracted
<code>extract [-a &lt;device&gt;] - &lt;file names&gt; ...</code>	Extract named files
<code>extract [-a &lt;device&gt;] -t</code>	List tape



**Do not enter the extract command without specifying a file name, because this command removes all files from the tape and fills up your disk. Use this command only under iNEWS supervision.**

This command takes programs not normally kept on disk from your release tape and adds them to your system. To extract programs from tape, load your release tape into the tape drive, select *all* servers, and type: `extract - <filename> <filename> ... <filename>` Wait until the tape stops before removing it from the drive.

```

force          force  -
(Superuser    force  [-q] [<name>] ...
only)         force  [-q] created>date1<date2 [<name>] ...
              force  [-q] lastlog>date1<date2 [<name>] ...
              force  [-q] passchg>date1<date2 [<name>] ...

```

This command forces users to change their password. For instance, to force user Harris to change her password, type: **force harris**

```

grpcheck      grpcheck [-v] <group story queue>

```

This command validates groups and aliases defined in each of the stories in the group directory (SYSTEM.GROUP by default). It then builds the alias file used by Avstar NRCS.

**Table A-16 grpcheck Command Options**

-v	Display processing status as grpcheck traverses the queue.
-vv	Display processing status as well as group, user, and alias statistics encountered.
-vvv	Display messages from -v and -vv, as well as the final list of groups and their members.

```

gtraits      gtraits  add  <group name>
(Superuser   gtraits  add  <group name>
only)

```

This command creates groups of users and modifies the security of existing groups.

The following lines show syntax for the gtraits command:

```

gtraits changegroup <pathname>
gtraits delete <group name>
gtraits interactive
gtraits list-
gtraits list [<group name>|<user name>]
gtraits rename <old group name> <new group name>
gtraits transfer <source group name> <destination
group name>

```

The first letter of each option can be used for shorthand.



help                    help <command name>

Displays information on how to use other commands. For instance, to get instructions for the `list` command, type:

**help list**

A message similar to the following appears:

```
usage:
  list configuration      [<termid> | <name>...]
  list directory         [<name>...]
  list terminal          [<termid> | <name>...]
  list user              [<name>...]
  list queue <name>     [<record limit>]
```

hogs                    hogs [<pathname>]

Scans the directories or queues you specify and displays usage information for them. This command is most useful when used on the People directory. For instance, to display usage information for the People directory, type:

**hogs people**

A display similar to the following appears:

%	USED	SHARED	HELD	LOCKED	PURGE	QUEUE NAME
0	136	20	0	0	0	PEOPLE.ARLIN

When you use this command, you are interested primarily in the number under `USED`. If this number is much higher for a certain queue than the numbers for other queues in the People directory, ask the queue's owner if the queue can be downsized.

`list`                    `list [mailbox=<# | name>] c [<terminal #> | <program name>] ...`

`list configuration`            [`<termid> | <name>...`]

`list directory`                [`<name>...`]

`list user`                    [`<name>...`]

`list group`                    [`<name | name>`]

`list queue`                    [`<name> [<record limit>]`]

`list ap=<number> d`

`list as=<number> d`

`list mailbox=<number | name> d`

`list ng=<group> d`

`list purge=<days>.[<hours>] d`

`list rg=<group> d`

`list rwg=<group> d`

`list rwng=<group> d`

`list save=[last | original | none | all] d`

`list sortfield=<field> d`

`list qform=<name>`

`list sform=<name>`

`list wg=<group> d`

`list a*`                    Lists all users whose login names begin with the letter “a.”

`list c`                    Lists current configuration of the system.

```
list d          list [<option>] d[- u | g | a | v | o] [<directory
name>] ...
```

**Table A-17 list d Command Options**

d-u	Lock user
d-g	Group information
d-a	Abstract printing traits
d-v	Verbose mode
d-o	Order user

Lists information about the specified directory or queue. If no directory or queue name follows `d`, the command displays information on the entire database.

For instance:

```
# list d dead
```

```
SRPlo-LIsUGQSXWFi sortfield  purge dis mbox directory
Q-R----I--G--X--- TITLE      P3.0 D1 -    DEAD
```

```
# list d-f dead
```

```
SRPlo-LIsUGQSXWFi queue form story form directory
Q-R----I--G--X---
DEAD
```

```
# list d-v dead
```

```
SRPlo-LIsUGQSXWFi sortfield  purge ap, al, as  dis mbox
DEAD:
Q-R----I--G--X--- TITLE      P3.0 A000,000,000 D1 -
      rg=-                wg=-                ng=-
      queue form=                story form=
```

```
list g          list g [<user or group name>] ...
```

Lists group information.

```
list q          list [qindex=<index value>] q[-a|b|d|g|m|s|v]
                <queue name> [<record limit>]
```

**Table A-18 list q Command Options**

q-a	Record address
q-b	Reverse order
q-d	Include deleted stories
q-g	Read-and-write group information
q-m	List who moved, duplicated, or killed the queue
q-s	Queue stamp
q-v	Verbose output

Lists information on the contents of a queue. For instance:

```
AVSTAR-B: list q people.palmer.new 1
```

A display similar to the following appears:

PEOPLE.PALMER.NEW id=126126						
rec	quick	index	LHDM-WObfpRmF	f.id	time	modified-time
25	h-disd		---M-----	13735	1	May2 17:07 2000

The index value consists of the selected sort field of the story you want to list. The quick index (qindex) value is optional, but must be a single word, and is not case-sensitive.

For instance, to get information for a story called “Nomad” in the queue PEOPLE.SMITH.NOTES, type:

```
list qindex=nomad q people.smith.notes
```

```
list s          list s [<session #> | <name>] ...
```

Lists session information, such as users currently logged in.

```
list u          list [<option>] u[- m | h | v | t | p] [<user or
group name>] ...
```

Lists user traits information, such as passwords, read rate, the keyboard description story, system setup and preferences, and mail and home queues.

If no name follows `u`, the command displays information about all users; otherwise, it displays information about the listed user, such as:

```
AVSTAR-A: list u danielmi
```

The result of the command will look something like this:

```
user      rr      kb  su  m   SOEKCVTH  sc  queues
danielmi  180    0   n   i   -OEKVTH  sc  dest: PEOPLE.D.DANIELMI.NOTES
                                     home: PEOPLE.D.DANIELMI
                                     mail: PEOPLE.D.DANIELMI.MAIL
AVSTAR-A:
```

The one-letter flags (`rr kb su m SOEKCVTH sc`) in the header provide current status information. The flags are:

<code>rr</code>	Readrate	<code>K</code>	Can Kill All
<code>kb</code>	Keyboard	<code>C</code>	Can Connect
<code>su</code>	Superuser	<code>V</code>	Can Video Browse
<code>m</code>	Insert/Overstrike Mode	<code>T</code>	Can Technical Direct
<code>S</code>	Simplified User	<code>H</code>	Can Highlight Read
<code>O</code>	Can Order	<code>s</code>	Can Configure Shortcut Toolbar
<code>E</code>	Can Enter & Remove	<code>c</code>	Can Configure Colors

The letters in the header are defined as follows:

**Table A-19 list u Command Options**

u-m	Mail queue
u-h	Home queue
u-v	Verbose output
u-t	Tracking information
u-p	Avstar for Windows preferences
<option>	List blacklist=[b] u
	List keyboard=<number> u
	List password= u
	List read rate=<number> u
	List su=[n] u
	List created>date1<date2 u
	List lastlog>date1<date2 u
	List passchg>date1<date2 u

logout

```
logout [<device #>] ...
logout all
```

Logs out a workstation. When you use `logout`, it saves the user's work before logging out his or her workstation. This command does not log out users in a connect session.

To log out specific workstations, follow the `logout` command with the device numbers of the workstations you want to log out. For instance, to log out workstations 12, 34, and 91, type:

```
logout 12 34 91
```

To log out all workstations, use `logout all`. Before logging users out, always broadcast a warning message and give them a chance to log out on their own.

`ls``ls [-l|a|c|d|F|l|q|r|R|s|t|u] [<directory name>`**Table A-20 ls Command Options**

<code>-l</code>	One entry per line
<code>-a</code>	All files
<code>-c</code>	By creation/modification time
<code>-d</code>	Only directory name
<code>-F</code>	Add “/” to directories and “*” to executable files in listing
<code>-l</code>	Long format
<code>-q</code>	Show non-printing characters as “?”
<code>-r</code>	Reverse order
<code>-R</code>	Recursively list subdirectories
<code>-s</code>	File size in blocks
<code>-t</code>	By modification date
<code>-u</code>	By access time

This UNIX command lists the contents of any UNIX directory. When using this command, you can choose the short form, which lists just the file names in the directory, or the long form, which lists names and characteristics of the files. For instance, to list the files in `/site/printers` using the long form, type:

```
ls -l /site/printers
```

A display similar to the following appears:

```
total 46
-rw-r--r-- 1 so 232 Mar 25 14:05 105
...
-rw-r--r-- 1 so 275 Mar 12 21:08 test3
```

From left to right, the display identifies file structure information, date and time the file was created, and file name.

To list file names in a particular directory, use `ls` without the `-l` option. Type:

```
ls /site/printers
```

A message similar to the following appears:

105	35	apcarbon	laser.bold	test1
115	45	apcarbonra	laser2	test3

This form just shows you the file names in the directory. (In this example, a few of the files' names are numbers, such as 105 and 115.)

`makeccutab`  
(Superuser  
only)

```
makeccutab -b[sv]
```

Builds new dictionary translations into the CCU program.

To use this command, stop all PCUs and CCUs, then become superuser and type:

```
makeccutab -b
```

**Table A-21** `makeccutab` Command Options

<code>s</code>	Standard translations
<code>v</code>	Verbose output

`makeshift`  
(Superuser  
only)

```
makeshift -[v|i|a|p|f|r <shift-file>] [file1 file2 ...]
```

Manages the case-shifting dictionary that Avstar uses to determine how to convert lowercase characters to their uppercase counterparts, and vice-versa.

Use the `makeshift` command in maintenance mode when you install Avstar to implement the case-shifting dictionary appropriate for the national language used at your site.



**Table A-22** **makeshift Command Options**

v	Verbose. To diagnose the case-shifting dictionary for potential errors, displaying messages for each line in the file. Checks that the file is readable and contains shift tables.
i	Install. To install the shift table into files you specify in the filename list.
a	Ask. To confirm installation of each file as you are installing the shift tables. Forces installation.
p	Print. To print shift tables contained in each file you specify in the filename list, with formatting similar to that in the default case-shifting dictionary. This option does not build or install the shift tables.
f	File. Specify with a <shift-file> filename, to use a file other than the default file for the case-shifting dictionary. The default file is (/site/dict/shift).
<shift-file>	When you specify the -f option, enter the case-shifting dictionary name you want to use instead of the /site/dict/shift file.
[<file1> <file2> ...]	When you specify the -i or -p option, enter one or more file names to install or print. If you specify a directory path instead of a file name, the <code>makeshift</code> program processes each file in the directory, then returns to the original directory.
[path]	When processing files in a directory, the <code>makeshift</code> program ignores additional directory pathnames it encounters, rather than recursively scanning child directories. To have <code>makeshift</code> scan all files in a directory, specify the directory path in the filename list.

`maketab`  
(Superuser  
only)

`maketab -b[sv]`

**Table A-23** **maketab Command Options**

s	Standard translations
v	Verbose output

Use this command after making changes to the dictionaries (or before the system is connected) to build the new translations into programs. To use this command, first stop all CCUs and PCUs. Then become a superuser and type:

**`maketab -b`**

mkdir  
(Superuser  
only)

mkdir <directory name>

This UNIX command creates a directory.

msgclean

msgclean -[r|d|o|t] [<username>|<number of days>

**Table A-24 msgclean Command Options**

-r	Remove invalid messages
-d	Show messages that are removed
-o	Show outstanding messages
-t	Tabulate outstanding messages
<username>	Specified user (use "*" for wildcard match)
#<days>	Older than number of days

If you remove a user who has unread messages, those messages remain in the database. You can remove them using the `msgclean` command followed by `-rd`. For instance, to remove the pending messages for all users that have been removed from the system, type:

```
msgclean -rd *
```

You can substitute the name of a removed user for the "\*" to remove only the messages for that user.

number

number <device number> ...

Causes a workstation to display its device number. To use it, you must log out the workstations that you want to display their device numbers. For instance, to find out which workstations have device numbers 51 and 52, type:

```
AVSTAR-A: logout 51 52
```

```
AVSTAR-A: number 51 52
```

After typing this command, find the workstations that have 51 and 52 in the upper-right corner of the screens next to the login prompt.

`offline``offline`

Puts Avstar NRCS offline. Users cannot log in, but users already on the system can continue normal function.

`online``online`

Puts Avstar online. Users can log in and use the system.

`otod``otod <number> ...`

**Table A-25 otod Command Options**

<code>leading 0</code>	Octal
<code>leading 0x</code>	Hex
<code>leading =</code>	Next characters (as characters)
<code>leading -</code>	Negative number
<code>leading _</code>	Two-character compose sequence

This UNIX utility command converts numbers from one base (such as decimal) to another (such as octal). For instance, to convert decimal 32, type:

```
otod 32
```

A message similar to the following appears:

```
h(0x20) o(040) d(32) u(32) SP,NUL,NUL,NUL Wed Dec 31
16:00:32 1969
```

In this listing, `h` stands for hexadecimal, `o` for octal, `d` for decimal, and `u` for unsigned binary. The number conversions are followed by the corresponding ASCII character (space, in this case), and the date value.

print            print <printer #> <command>

Redirects the output from any console command into a story in the Dead queue and prints it.

For instance, to print the configuration file on printer 1, type:

**print 1 cat /site/config**

query            query <ccu #> | <mcs pc devid>  
query <device #> <qtype> [parameters]

Syntax:

<devid> ioregister  
<devid> iocontrol  
<devid> proc  
<devid> input  
<devid> output  
<devid> read <addr> <length>  
<devid> absread <seg> <addr> <length>  
<devid> writeb <addr> <byte value in hex>  
<devid> writew <addr> <word value in hex>  
<devid> abswrite <seg> <addr> <word value in hex>  
<devid> netstat  
<devid> bstat

Tests communication between a CCU, PCU, or MCS-PC and its server.

Type **query** followed by the device number of the CCU, PCU, or MCS-PC. For instance, to test CCU 20, type:

AVSTAR-A: **ccureset 20**

A message similar to the following appears if the CCU, PCU or MCS-PC is functioning properly:

```
#@> 14.01.03 CCU CCUDL OK
```

If the system returns an OK message a few seconds after you execute the command, the CCU, PCU, or MCS-PC is functioning properly. If

there is a problem, the server displays a failed message after about a minute.

```
reconnect <name> [<option>=<value>] ...
```

The options for this command are the same as for the `connect` command.



**Enter the correct identifier, such as A, B, C, or D, for the host computer (server). Reconnecting a host computer that is currently connected disrupts operation of all CCUs and PCUs configured on that machine. For instance, entering `reconnect a` when the A server is already connected causes all CCUs and PCUs configured on the A server to fail.**

Connects a server to a running Avstar. You must type the `diskclear` command first on the server being reconnected before reconnecting it to the system.

```
rename  
(Superuser  
only)
```

```
rename [- u|v|r] <old name> <new name>t
```

**Table A-26 rename Command Options**

-u	Update user record only, changing mail, destination, and home directory entries that match <old name>
-v	Verbose output
-r	Resume interrupted operation



**Resuming interrupted operations after changing queue or directory names could cause loss of data.**

Renames any directory or queue in the database. You must become superuser, take the system offline, and log out all users before using `rename`. When you use this command, the system must be named and offline. For instance, to rename the People directory to `PEOPLE.STAFF`, type:

```
rename people people.staff
```

The command creates any new directory levels that are necessary.

restart restart [-v] <device> | <device all>



**If you start a device when a user is editing, data could be lost.**

Restarts one or more devices, an entire CCU or PCU (and its devices), or every CCU and PCU and their devices on a server. The `restart` command stops and reloads the currently executing program(s).

To restart a device, type **restart** followed by the device number. For instance, to restart printer 41, type:

```
restart 41
```

To restart all devices on a CCU or PCU, type **restart** followed by the CCU's or PCU's device number. To restart all devices on your system, type:

```
restart all.
```

After each restart, you see a `Hot-to-go` message for each device as it starts. If the device does not start, you see a message indicating that the restart of that device has failed.

rmdir rmdir <directory name>



**Once the specified file is removed, you can recover it only by restoring a backup to that system. This UNIX command removes an empty UNIX directory. Use this command only if instructed to do so by iNews personnel.**

```
searchtape [on <device>] [from <date> to <date>]
[max <# stories>] for <word> ...
searchtape [on <device>] [just <date>]
[max <# stories>] for <word> ...
```

Use one of the following date formats: YY, YYMM, or YYMMDD.

Searches a tape created by the `dbdump` command and recovers stories from it. Stories that contain a word specified in your search are restored to the queue `SYSTEM.SEARCHTAPE`. For instance, to search a tape for stories containing the word “dinosaur,” type:

```
searchtape for dinosaur
```

A message similar to the following appears:

```
8 stories restored to SYSTEM.SEARCHTAPE
```

In this case, `searchtape` found eight stories containing the word “dinosaur” and placed them in `SYSTEM.SEARCHTAPE`.

When searching a tape, you can specify a date range when the story was saved, as well as the maximum number of stories for the system to restore.

```
send          send <username> <message>
```

Lets you send messages to users from the console or from any VT/DOS terminal.

Here are two examples of messages sent to user Smith:

```
send smith are you editing ap-arson
send smith please log out now
```

```
shutdown     shutdown
```



**Typing this command while users are editing may cause data loss.**

Stops all devices and closes the database. It is used before halting the system. To use this command, you must first select *all* servers and log everyone off the system.

```
sitedump     sitedump [<device>]
(Superuser  Makes backups of your system's site files.
only)
```

siterestore            siterestore [<device>]  
(Superuser  
only)



**All unsaved changes to site files will be lost, and any unsaved site files will be replaced by the version on the backup tape. You will lose the version currently on your system.**

Restores site files and programs backed up to tape with `sitedump`.



*After performing a siterestore, type the `grpcheck` command to rebuild the mail aliases file.*

softdump            softdump [<device>]

Dumps a complete image of the software area of the disk to tape.

softrestore        softrestore [<device>]



**This function must be performed from single-user mode. Restoring an old or inaccurate version of the data may cause functionality problems.**

Use this command on an SCO UNIX system only to restore the software area of the disk from a `softdump` tape. Use it only when advised to do so by iNews personnel.

startup            startup

Starts the system's devices after they have been shut down. The system must be offline and all devices must be stopped.



```
status          status [all | license]
```

Displays system connection information: which server is the master, if system is running single or dual, and the disk status.

There are three disk statuses:

- **OK** - When the system is up and running either as dual or single.
- **Cleared** - When you have cleared the database of a failed server.
- **Unknown** - When you reconnect the CPUs following a `diskclear`. When the `diskcopy` procedure has completed and the database has been mirrored, the disk status will change back to OK.

To list basic system information, type:

```
status
```

A message similar to the following appears:

```
A is online  ID is AVSTAR  
System is AB. Master is A.  
Disk status is UNKNOWN
```

To list system options set in the system profile, type:

```
status all
```

To list the system's license information, type:

```
status license
```

To list basic system information, including maximum number of devices allowed on your system, type:

```
status |
```

stop                   stop [-v] [all | <device number>]



**If you stop a device when a user is editing on the device, data could be lost.**

Stops activity on a server prior to shutting it down. Before using the `stop` command to stop an activity, use the `broadcast` command to notify users the system will be going down, and log out all devices on the affected servers. To stop all devices on a server, use `stop all`.

To stop a device, follow `stop` with the number of the device you want to stop. For instance, to stop workstation 12, type:

```
stop 12
```

The `stop all` command differs from the `shutdown` command in that the free list remains in memory and is not flushed back to the disk.

su                   su

This UNIX command makes you a superuser. To become a superuser, type **su**, then type the superuser password when prompted.

The display looks similar to the following:

```
AVSTAR-A: su
password:
SU: so /dev/console
```

unbusy               unbusy <queue name>

Removes edit and order locks from the specified queue in your database.



**If a user is actually working in the file, removing the lock could cause data loss.**

utraits  
(Superuser  
only)

utraits <username> [<option> <value>] [+|- flag]

**Table A-27 utraits Command Options**

destination	realname
keyboard	home
readrate	password
editmode	su
mail	

**Table A-28 utraits Command Flags**

b	Blacklist	vb	Video browse
c	Connect	td	Technical direct
er	Enter and remove	hr	Highlight read
ka	Kill all	cc	Configure color
o	Order	cs	Configure shortcut toolbar

name

name: <username>  
<string>\*  
all

You can enclose the real name in double quotation marks to allow for embedded spaces.

Sets a user's traits. For instance, to assign keyboard 12 to user Jones, type:

**utraits jones keyboard 12**

**Table A-29 name Command Options**

destination	editmode
home	keyboard
mail	password
readrate	su

**Table A-30 name Command Flags**

<b>b</b>	Blacklist
<b>name &lt;username&gt;</b>	Modify user with the name <username>
<b>&lt;string&gt;*</b>	Modify users with names starting with <string> * indicates wildcard. For instance, "a*" modifies all users with names starting with "a".
<b>all</b>	Modify all users

`version`

```
version [-<alternate pattern>] <filename> ...
version [+]
```

Displays the version and platform of the Avstar software you are using. Type:

**version**

A message similar to the following appears and displays product version information:

```
version: 14.01.05 (C) DECsystems
```

You can also use this command to display a particular program's version number. Type **version** followed by the program's name. For instance, to find out which version of the `dbsort` program you are using, type:

**version dbsort**

A message similar to the following appears:

```
dbsort: 14.01.05 (C) DECsystems
```

`wholockedit`

```
wholockedit <queue name> [all]
```

Displays who locked a story. For instance, to find out who locked a story in `PEOPLE.SMITH`, type:

**wholockedit people.smith**

To find out who last modified each story in a queue, type **wholockedit** followed by the name of the queue and the keyword **all**.

wiredump

```
wiredump <device number> <speed> <bits>
-<code[rotate]> <style> / <file> <count>
```

<bits>	Bits per character (5, 6, 7, or 8) or speed options, as interpreted by the <code>ccuspeed</code> program.
<file>	File name for collected data
<count>	Number of bytes of data to collect
<code>	b for Baudot, t for TTS, a for ASCII, w for Wire ASCII, or e for 8-bit ASCII
[rotate]	1-7 to rotate right <x> bits; 8 for one's complement (010=>101); 9 for inversion (123=>321)

If a problem exists with your wires, you may need to use this command to collect raw wire data so it can be sent to Avstar for testing. The `wiredump` command can also be used for debugging format changes made by a wire service. A wire dump can be sent to the screen or to a disk file.

For instance, if device 35 is a 1200 baud, ASCII, 7-bit wire, type the following to print the first 200 characters to the screen:

```
wiredump 35 1200 7 -a c
```

To dump wire data to a file, use the following format. The file name refers to the output file into which the `wiredump` data is placed. It should include the name of the wire and be preceded by the `/tmp` directory name. The count is measured in number of bytes.

For instance, if device 35 is the APTV wire, to display the first 2000 characters, type:

```
wiredump 35 1200 /tmp/aptv.raw 2000
```

A display similar to the following appears:

```
loadtime = 0:11
AVSTAR-A: restart 35
W35: 14:34:09 Hot-to-go
```

## Job List Commands

The following section provides a list of commands that can be used in a job list, which is created and modified in the database. The command's format and description are provided, followed by a list of servers that can utilize the command in their job lists.

`at`                    `at <hh:mm>`

Specifies the time of day when a task will take place. You can combine this instruction with the keyword `on` to specify both the day and time for the task.

Applies to action and tx servers.

`bscan`                `bscan <queue name> [priority | all | everyentry]`

Works like `scan`, except it reads stories in the scan queue in reverse chronological order, reading the newest stories first. This is convenient for very large queues.

Applies to action and tx servers.

The `priority` option forces the action/txnet server to interrupt the scan of another queue if the action/txnet server receives a mailbox notification. The `all` option forces the action/txnet server to scan the entire queue instead of the limit of 10 stories. The `everyentry` option forces the server to process each entry in a queue, not just modified entries.

---

distribution	<code>distribution &lt;distribution story queue&gt; [&lt;error queue&gt;]</code> <p>Specifies the queue containing the distribution story and, optionally, an error queue for stories whose distribution codes cannot be processed.</p> <p>Applies to distribution and parallel servers.</p>
dup	<code>dup &lt;destination queue&gt; [&lt;distribution code&gt;]</code> <p>Copies the stories in the scan queue to a queue you specify, optionally including a distribution code with them.</p> <p>Applies to action, tx, and serial rx servers.</p>
every	<code>every &lt;dd:hh&gt;</code> <p>Specifies the interval at which a task is performed. You can set this value in days and/or hours.</p> <p>Applies to action and tx servers.</p>
fullform	<code>fullform</code> <p>Instructs the Tx link to transmit the full form text of each story, rather than just the story's form name.</p> <p>Applies to tx servers.</p>

<code>ignore</code>	<code>ignore [yes   no]</code> <p>Including <code>ignore yes</code> in a job list that performs validation ensures the server or link accepts any values for the fields it is validating. The default is <code>ignore no</code>.</p> <p>Applies to action and tx servers.</p>
<code>ignore-del</code>	<code>ignore-del</code> <p>Causes a server or link to take no action when a story is deleted from its scan queue.</p> <p>Applies to action and tx servers.</p>
<code>local</code>	<code>local &lt;queue name&gt;</code> <p>Specifies the primary wire queue for a parallel wire server.</p> <p>Applies to parallel servers.</p>
<code>move</code>	<code>move &lt;destination queue&gt; [&lt;distribution code&gt;]</code> <p>Moves stories from the scan queue to a queue you specify, optionally adding a distribution code to them. It must be the last instruction in a job list task.</p> <p>Applies to action and tx servers.</p>
<code>number</code>	<code>number &lt;form field&gt; &lt;length&gt; [&lt;error queue&gt;]</code> <p>Assigns a unique number to each story in the scan queue. Specify the form field that will display the number and the number of digits displayed.</p> <p>Applies to action and tx servers.</p>



---

on	on <days>	Indicates on which days of the week a time-interval task will occur. You can combine this instruction with the <code>at</code> keyword to indicate both day and time.  Applies to action and tx servers.
open	open <computer> <username> <format> [<queue name>] <story name>	Initiates a network connection with a remote system for story transfer. The <code>username</code> you specify must exist with identical passwords on both the local and remote systems.  Applies to network and tx servers.
order	order	Indicates that order changes in a Tx link's scan queue should be transmitted to the remote system. For this to work correctly, the destination queue on the remote system must have its update trait turned on.  Applies to tx servers.
publish	publish [no yes]	When placed following a <code>scan</code> or <code>bscan</code> line, the <code>txnet</code> publishing option, <code>publish no</code> , disables sending <code>published.html</code> when using HTML export. The default is set to <code>yes</code> .
put	put <queue name>	Sends stories over a network Tx link to a specified queue on the remote system.  Applies to network and tx servers.

quiet	quiet [no yes]
	Including <code>quiet no</code> in a job list that performs validation sends a message whenever the server or link successfully validates a story. The default, <code>quiet yes</code> , means a message is sent only when a story fails validation.
	Applies to tx servers.
remote	remote <queue name>
	Identifies the secondary wire queue scanned by a parallel wire server.
	Applies to parallel wire servers.
remove	remove
	Deletes stories from the scan queue. It must be the last instruction in a job list task.
	Applies to action and tx servers.
replace	replace <destination queue name>
	Works like the <code>dup</code> command, except that it updates stories in the destination queue only when they are changed in scan queue. It does not copy new stories from scan queue to destination queue.
	Applies to action, tx, and serial rx servers.
scan	scan <queue name> [priority   all   everyentry]
	Specifies queue monitored by this task. The scan line must come before any instructions that manipulate stories in the queue, like <code>dup</code> or <code>move</code> . Ten stories are scanned at a time from each scan queue; adding <code>priority</code> to a scan line means all new or modified stories in that queue are scanned at once. The queue identified in the <code>scan</code> command as the priority queue is always the next queue in the multiple-scan job list, so if it is idle, other queues are processed. The system

checks after every queue to see if new stories are ready for processing on the queue identified in the `scan` command.

Applies to the action and tx servers.

The `priority` option forces the action/txnet server to interrupt the scan of another queue if the action/txnet server receives a mailbox notification. The `all` option forces the action/txnet server to scan the entire queue instead of the limit of 10 stories. The `everyentry` option forces the server to process each entry in a queue, not just modified entries.



*Deleted entries are still controlled by the `send-del` option.*

`send-del`

`send-del`

Instructs the server or link to process story deletions in the scan queue; this is the default behavior. Use `ignore-del` to have the server or link take no action when a story is deleted from a scan queue.

Applies to action and tx servers.

`source`

`source <queue name>`

Specifies a queue that a distribution or keyword server should check each time it wakes up. Each task in a job list for such a server must begin with a source line.

Applies to the distribution and keyword servers.

`validate`

`validate <validation queue> <error queue>`

Activates form field validation for a server or link. It must include the queue name containing the validation story and an error queue for stories that cannot be validated.

Applies to action and tx servers.

## Dialog Commands

This section describes the dialog commands. Some of these commands are equivalent to those available to a user during a connect session, while others are unique to dialogs.

Each command must begin on a new line and can be uppercase or lowercase. The system does not check dialogs for errors.

Many of these commands take one or more strings as their parameters. These strings are text and can be entered from the keyboard or, if you need to include characters not available from your keyboard, use their aliases or decimal equivalents. This applies to nonprinting characters, such as Escape and Enter, which are represented by <esc> and <cr> respectively.

Depending on which character set is used by the device to which you connect, you may need to translate certain keyboard characters to characters understood by the device. You may also need to translate some of the characters sent by the device to characters usable by your system.

Translating characters is called *mapping*, and there are three commands (`map`, `mapin`, and `mapout`) that allow you to do this. While these commands are also available to users, system administrators will usually use them in dialogs to set up the translations for users rather than leave it for users to do after they have connected.

`capture`

`capture <destination queue name>`

Places a copy of all text received from the remote connection during a session in a story into the queue you specify.

Usually you invoke `capture` from the `cmd>` prompt. However, to turn `capture` on in a dialog, place the `capture` command and destination queue name at the point in the dialog where you want to begin capturing material.

You must include a destination queue unless you are restarting `capture` after having paused it. If you have not paused `capture` earlier

in the dialog, leaving out the queue name generates an error and terminates the dialog.

`delay`                    `delay <# of seconds>`

Pauses the dialog for a number of seconds.

When the specified time has passed, the dialog resumes. Put the command where you want the dialog to pause and follow it with the number of seconds you want the dialog to pause. For instance, to pause the dialog for five seconds, type `delay 5`.

Although the dialog is suspended while this command is in effect, you can use the `quit connect` command to close the connection.

`diag`                    `diag [on | off]`

Normally, a dialog's diagnostic mode is off and screen output is suppressed while the dialog is running. However, you can use the `diag` command to turn the dialog's diagnostic mode on so you can see what the dialog is doing as it executes.

Usually you want the diagnostic mode on only when you are debugging a dialog, so you can determine exactly where any errors occur. Place a `diag on` command in the dialog at the point where you want to start debugging, as shown below.

To see what happens during just one part of a dialog, bracket that portion of the dialog with `diag on` and `diag off` commands.

`echo`                    `echo [on | off]`

Turns local echo on or off.

Turn on local echo in any dialog used to connect to a device or information service that does not echo back what the user enters. This way, the user can see what he or she is entering.

To turn on local echo, place `echo on` in the dialog where you want local echo turned on. Use this command at or near the beginning of the dialog.

While you can turn local echo off using the `echo off` command, you are not required to do so. Local echo is turned off automatically when the dialog finishes.

`escape` `escape <escape character>`

To change the escape character (used to bring up the `cmd>` prompt) from within a dialog, use the `escape` command.

For instance, to change the escape character from the default CTRL-] to CTRL-Z, use the `escape` command. Represent the CTRL-Z character as `<26>` (its decimal value).



*Do not change the escape character to CTRL-A, CTRL-Q, or CTRL-S. These characters have other important functions.*

The escape character is reset to the default (CTRL-]) when the user closes the connection.

`expect` `expect`  
`<delimiter><string1><delimiter><string2><delimiter>...`

Instructs the dialog to wait for the device to which the service has connected to send a string (*string1*). If that string is not received within five seconds, `expect` sends the second string (*string2*) to the device and waits for a third string (*string3*). This continues until an expected string is received within the five-second limit or `expect` runs out of strings.

To use the `expect` command, follow it with the character—that is, `<delimiter>`—you want to use as the delimiter to separate each string in the list. The delimiter can be any character. Follow this with the first string you want `expect` to wait for. Then add a delimiter and the string you want `expect` to send if it does not receive the first string. You can add as many strings as you want.

For instance, some devices may not display a `login:` prompt unless you press Enter. To have the dialog send a carriage return to the device if it does not receive the `login:` prompt immediately, use the `expect` command. The first character following `expect` in this example is a comma. This sets the comma as the delimiter used to separate strings following the command from each other.

If you do not place a string between two delimiters, this indicates a null string. If you have the `expect` command wait for a null string, it considers any string it receives to be a match. If you have the `expect` send a null string, it does not send anything, but instead waits for the next expected string.

Although the dialog is suspended while this command is in effect, the user can employ the `quit connect` command to close the connection.

`heol`

`heol [on|off]`

If necessary, you can have the system insert a hard end-of-line character (HEOL) after each line of captured text.

Put the `heol on` command in the dialog at the point where you want this feature turned on.

To turn off this feature, use `heol off`.

If you are calling a device that contains information in tables or columns, have the system insert an HEOL at the end of each captured line if the device from which you are capturing the material does not. This way, tables and columns you capture retain their format, even if someone packs the story. Put the `heol on` command in the dialog.

`map`

`map <local character> <remote character>`

Translates a character (local character) entered at the keyboard to some other character (remote character) before sending it to the device to which you are connected. Likewise, if the system receives a remote character from the device, the `map` command translates it to local character before sending it to a workstation.

You may use `map`, for instance, if you created a service that opened a UNIX port on a system so your workstation can act as a console on that system. While connected, you may see the UNIX pipe, “|”, displayed on your screen, or have occasion to enter this character, or both.

However, this character indicates the beginning of a graphics command for your workstation, and typing it at the keyboard or displaying it on the screen confuses your workstation. So, you must map the “|” character to another character that your workstation can accept, such as the exclamation mark (!). Use the `map` command to accomplish this. Use the decimal equivalent (<124>) for the “|” character, since you cannot enter that character at an Avstar Workstation.

Used this way, `map` translates every exclamation point (!) you typed to a “|” before sending it to the device you have connected to. Likewise, every “|” character the device sends to your workstation is translated to an exclamation point (!) before it is displayed on your screen.

`mapin`                    `mapin` <remote character> <local character>

Translates a specific character (remote character) received from the device to which you are connected to another character (local character).

For instance, if the device to which you are connected sends the “|” character and you want it translated to an exclamation point (!), use the `mapin` command. Again, use the decimal equivalent (<124>) for the “|” character.

This translation affects only characters received from the device to which you are connected. It has no effect on the character when you enter it.

`mapout`                    `mapout` <local character> <remote character>

Translates a particular character entered at the workstation (local character) to another character (remote character) before it is sent to the device to which you are connected.



For instance, some devices use a limited character set that does not recognize lowercase letters. To connect to such a device, you would want to map all lowercase characters to their uppercase equivalents. For instance, to map a to A, use this `mapout` command:

**`mapout a A`**

This has no effect on characters received from the device. Only characters typed at the workstation are translated.

`message`

`message <string>`

Sends a string to the screen when you have typed the `echo off` command.

This command informs users that the system is active and functioning during a dialing session.

`pass`

`pass <resume character>`

Suspends a dialog and yields control to the user.

Whatever the user enters after the `pass` command is executed is sent directly to the device to which the user is connected. When the user enters the character defined in `<resume character>` parameter, `pass` resumes the dialog.

To use `pass`, place it in the dialog where you want to yield control to the user and follow it with a character you want the user to enter to resume the dialog. When the user enters this character, `pass` sends it to the device and then resumes the dialog, preventing further user input.

For instance, suppose you have a dialog that logs you into an information service. For security reasons, you want the dialog to pause at the password prompt, let you enter the password, and then resume. Do this using the `pass` command followed by a `<cr>` so pressing Enter after entering the password resumes the dialog.

If you use `pass` without a parameter, it passes everything the user enters until he or she tries to close the connection with the `quit` com-

mand. Then the dialog resumes, executes the remaining commands in the dialog, and closes the connection.

pause

pause

Suspends capturing from within the dialog. If you turned `capture` on earlier in the dialog, you can pause capturing using the `pause` command.

To resume capturing later in the dialog, include a `capture` command (without a destination queue) at that point in the dialog.

stop

stop

Stops capturing from within a dialog.

If you have turned `capture` on earlier in the dialog, you can turn it off using the `stop` command.

timer

timer <# of seconds> <string to display>

Counts number of seconds specified in <# of seconds> when it is activated by the next `wait` command in the dialog.

When a specified string in the `wait` command is received, `timer` stops counting and `wait` resumes the dialog. If `wait` does not receive expected string within seconds as specified in the `timer` command, `timer` displays the text specified in string and closes the connect session.

To use `timer`, follow it with the number of seconds you want it to count and the string you want it to display if that period of time elapses. For instance, you may want to use the `timer` command so it terminates the session if the dialog is unable to log in within 60 seconds.

When a `pass` command is active, an active `timer` command suspends counting. When `pass` command finishes, `timer` command resumes counting.

Also, the same `timer` command applies to any subsequent `wait` commands if no other `timer` commands appear before them. If you do not want to use the same `timer` value for another `wait` command later in the dialog, insert `timer 0` after the first `wait` command. This cancels the first `timer` command and causes subsequent `wait` commands to wait for their string forever if no other `timer` commands follow `timer 0`.

`type`                    `type <string to send>`

Sends a string to the device to which the service has connected.

For instance, if you were creating a dialog that types the user's name in response to a login prompt, you may use:

```
type joel smith
```

Most devices to which you connect expect a carriage return (represented by a `<cr>`) after each string you send. When this is the case, you must include a `<cr>` at the end of the string, as shown in the example.

`wait`                    `wait <string to wait for>`

Pauses the dialog until a specified string is received from a device to which the service has connected, or until a certain amount of time (specified by a `timer` command) has elapsed with no response.

To use this command, follow it with the string for which the dialog should wait.

Unless a `timer` command has been executed first, the `wait` command waits forever until the specified string is received, so type exactly the string you want it to wait for, and keep in mind that `wait` is case-sensitive. If the dialog never receives the string `wait` is looking for, the dialog hangs, and you need to use the `quit connect` command to exit the dialog and return to your Avstar Workstation.

If you use `wait` without a parameter, the dialog waits until any character is received.



# APPENDIX B

## *System Files*

This appendix contains samples of the system files you are most likely to reference or change:

- `/etc/bootptab`
- `/etc/hosts`
- `/etc/networks`
- `/site/config`
- `/site/printers/ti830`
- `/site/system`
- `/site/wires/anpa7`
- `console.cfg`
- `SYSTEM.CLIENT.WINDOWS`
- `SYSTEM.CONFIGURE.301-ACTION`
- `SYSTEM.MAP`
- `SYSTEM.RESOURCE`
- `SYSTEM.WIRES.DISTRIBUTION`
- `SYSTEM.WIRES.KEYWORDS`
- `SYSTEM.WIRES.KEYWORDS-AP`
- `SYSTEM.WIRES.KEYWORDS-AP2`

## /etc/bootptab

### SCO-UNIX Specific

```
#
#      @(#)bootptab      4.3 Lachman System V STREAMS TCP  source
#      SCCS IDENTIFICATION
# /etc/bootptab: database for bootp server (/etc/bootpd)
# Blank lines and lines beginning with '#' are ignored.
#
# Legend:
#
#      first field -- hostname
#                      (may be full domain name and probably should be)
#
#      hd -- home directory
#      bf -- bootfile
#      cs -- cookie servers
#      ds -- domain name servers
#      gw -- gateways
#      ha -- hardware address
#      ht -- hardware type
#      im -- impress servers
#      ip -- host IP address
#      lg -- log servers
#      lp -- LPR servers
#      ns -- IEN-116 name servers
#      rl -- resource location protocol servers
#      sm -- subnet mask
#      tc -- template host (points to similar host entry)
#      to -- time offset (seconds)
#      ts -- time servers
#
# Be careful about including backslashes where they're needed.  Weird (bad)
# things can happen when a backslash is omitted where one is intended.
#
# First, we define a global entry which specifies the information every host
# uses.
#
global.dummy:\
    :sm=255.255.255.0:\
    :hd=/usr/boot:bf=null:\
    :ds=128.2.35.50 128.2.13.21:\
    :ns=0x80020b4d 0x80020ffd:\
    :ts=0x80020b4d 0x80020ffd:\
    :to=18000:
#
# Next, we can define different master entries for each subnet. . .
```

```

#
subnet13.dummy:\
    :tc=global.dummy:gw=128.2.13.1:
subnet232.dummy:\
    :tc=global.dummy:gw=128.2.232.1:
subnet19.dummy:\
    :tc=global.dummy:gw=128.2.19.1:
#
#
pcu20: ht=1:    ha=0020af48001b:    ip=152.165.17.101:
pcu30: ht=1:    ha=02608cda7e0e:    ip=152.165.17.102:
#pcu40: ht=1:    ha=02608cdbfa02:    ip=152.165.17.103:
#pcu50: ht=1:    ha=02608cdbcf32:    ip=152.165.17.104:
ap-pcu: ht=1:    ha=02608c123456:    ip=152.165.154.22
#
# END

```

### SGI-Irix Specific

```

# @(#) news/release/irix/etc/bootptab 14.1.6.1      Date 99/07/07
#
# /usr/etc/bootptab:  config file for bootp server
#
# Boot directory
/exc/ccu/at
#
# default bootfile
pcuos.exe
#
# end of first section
#
%%
#
# The remainder of this file contains one line per host
# with the information shown by the table headings below.
#
# Note that bootfile MUST exist in the boot directory
# specified above for the server to respond to requests.
#

# host  htype  haddr                iaddr                bootfile
pcu10   1        02:60:8c:12:34:56    125.1.10.1           pcuos.exe
pcu20   1        02:60:8c:12:34:57    125.1.10.2           pcuos.exe
pcu30   1        02:60:8c:12:34:58    125.1.10.3           pcuos.exe
pcu40   1        02:60:8c:12:34:59    125.1.10.4           pcuos.exe

```

**DEC/MIPS Specific**

```

#
# @(#) news/release/com/mvax/etc/bootptab      14.1.3.2      Date 99/07/21
#
# Address table for the bootp server (bootpd).
#
# This first section is unused but must be present.
#
/tmp
defaultboot
%%
#
# This section maps ethernet addresses to IP addresses.
# host      htype  Ethernet address  IP address  bootfile
#          (always 1) (use :05: not :5:)                (unused, must be present)
#
toranaga   1      31:5e:54:62:55:67  125.1.0.33  defaultboot
taipan    1      04:07:11:20:35:46  125.1.0.76  defaultboot
pcu160    1      02608c:dd8714     124.1.100.160 defaultboot
pcu30     1      02608c:dd869c     124.1.100.30 defaultboot
line      16    1200-7e modem hayes ;[1-02] modem
line      17    1200-7e modem hayes ;[4-04] TI880
;
pcu 20   pcu20  at 21 22 23 24 25 26 27 28
workstation 21 9600-7e 2 news - ;[1-02] 6p

```

**/etc/hosts**

```

# @(#)hosts      1.2 Lachman System V STREAMS TCP  source
#      SCCS IDENTIFICATION
127.0.0.1      localhost
125.1.0.1      AVSTAR-A A avstar-a a
125.1.0.2      AVSTAR-B avstar-b.local avstar-b B b
125.1.0.8      ARCHIVE2 A2 a2 archive2
125.1.10.10    pcu10  02608c0ebfc9
125.1.10.20    pcu20  02608c1aa64f
125.1.10.30    ccu30  02608cd8d8ed
125.1.10.40    ccu40  02608cd8d8dd

```



## /etc/networks



*This file may vary by platform, particularly in network device name, that is, In0, e3B0, ec0, and so forth.*

```
# Network name database (see networks(4) for more
# information).
loopback          127
#
# Internet networks
#
mirror            124          ec0
device            152.165        ec1
```

## /site/config

```

;HOST SECTION
host      ab      a
net       n125    20 300
reslist   100 106 108 200 202 204 250 252 254 400
servers   120 122 124 126 128 140 142
;
host      ab      b
          net      n125    10 30 301
          reslist   105 107 109 201 203 205 251 253 255 401
          servers   121 123 125 141
;
; ALTERNATE HOST SECTION
;
host      a      a
          net      n125    10 20 30 300 301
          reslist   100 106 108 200 202 204 250 252 254 400
          reslist   105 107 109 201 203 205 251 253 255 401
          servers   120 122 124 126 128 140 142
          servers   121 123 125 141
;
host      b      b
          net      n125    10 20 30 300 301
          reslist   100 106 108 200 202 204 250 252 254 400
          reslist   105 107 109 201 203 205 251 253 255 401
          servers   120 122 124 126 128 140 142
          servers   121 123 125 141
;
; DEVICE/RESOURCE/SERVER SECTION - BODY
;
ccu       10      ccu10      at      11 12 13 14 15 16 17 18
;

printer   11      2400-8n  2      -      ; LaserJet 4
terminal  12      19200   0      news   -      ; edit rm 3
terminal  13      9600-8n  1      news   -      ;
wire      14      2400     anpa7  RT    -      ; Reuters
terminal  15      9600     1      news   -      ;
dialup    16      38400-7ea 0      news   D16    ;
printer   17      9600-8n  1      ; script
wire      18      1200-8n  dummy  A2    -      ; backup AP
;
ccu       20      pcu20     pc      21 22 23 24 25 26 27 28
;
wire      21      9600-8n  dummy  DD    -      ;
terminal  22      9600-8n  1      news   -      ;
terminal  23      19200-8n  1      news   -      ;
terminal  24      19200-8n  0      news   -      ;

```

```

line          25    9600-8na      modem      hayes      ; dialout
dialup        26    38400-7ea     0 news      -          ;
terminal      27    19200-8n     1 news      -          ;
unused        28                                ; bad port
;
ccu           30    pcu30        pc         31 32 33 34 35 36 - -
;
line          31    9600-8n     modem      hayes      ;dialout
wire          32    9600-8n     anpa7 AP   -          ;AP wire
terminal      33    19200-8n     1 news     -          ;
terminal      34    19200-8n     1 news     -          ;
terminal      35    19200-8n     1 news     -          ;
dialup        36    38400-7ea     0 news     D36       ;
;
;
anws          200    125.1.100.85 0 gnews     -          ; News Director
anws          201    -            1 gnews     -          ; anws
anws          202    -            1 gnews     -          ; anws
anws          203    -            1 gnews     -          ; anws
anws          204    -            1 gnews     -          ; anws
anws          205    -            1 gnews     -          ; anws
andos         250    -            2 news      -          ; andos
andos         251    -            1 news      -          ; andos
andos         252    -            1 news      -          ; andos
andos         253    -            1 news      -          ; andos
andos         254    125.1.100.83 1 news     -          ; hansen
;
terminal      100    0            1 news     -          ; net term
resource      105    console      -          -          ;
resource      106    archive      -          -          ;
resource      107    net          -          -          ;
special       108    0 txnet      108       -          ; txnet
special       109    0 rxnet      -          -          ; rxnet
;
server        120    mailserver    120       -          ; mailserver
server        121    printserver   121       -          ; slave printing
server        122    seek          122       -          ; seek server
server        123    keyword       123       kwd        ; kwdserver
server        124    action        124       -          ; timed-action
server        125    action        125       -          ; action server
server        140    monitor       140       -          ; 6 pm rundown
server        141    monitor       141       -          ; 11 pm rundown
server        142    monitor       142       -          ; show.cnn.rundown
;
mcspc         300    mcspc1       avidapdriver ap1        -          ; airplay
mcspc         301    mcspc2       avidapdriver ap2        -          ; airplay 2
;
websession    400
websession    401

```

## /site/printers/ti830

```

; @(#) news/release/com/printers/ti830 14.1.4.1          Date 00/08/06
;
;TI830 printer profile 05/09/00
;
initprint          <esc><cr>P
ejectcode          <ff>
ejectcount         1
idlecount          0
scriptrhstart     30
scriptrhmax       24
scriptlhmax       24
scriptshift       yes
scripttemplate    no
;
expand            <SO><SI>          <DC4><DC2>  ;double-wide, compressed
;
font 1            <esc>Q           <esc>R           ;emphasized: better normal
font 2            <esc>Q<esc>K3    <esc>R<esc>M      ;enhanced + emph: newscaster
font 3            <esc>K3         <esc>M           ;enhanced: video
font 4            <esc>I          <esc>J           ;underline
font 5            <esc>I<esc>K3    <esc>J<esc>M      ;enhanced + undrl: director
;
form 1            <esc>d<esc>y<esc>f0 <nul>           ;draft, 10cpi, courier
form 2            <esc>q           <esc>d<esc>y      ;quality, then draft
form 3            <esc>f1         <esc>f0          ;sans serif, then courier
form 4            <esc>f2         <esc>f0          ;italics, then courier
form 5            <esc>f3         <esc>f0          ;sans serif/italics, then courier
...

```

## /site/system

```

id=AVSTAR net=ab
textmax=80          scriptlhmax=25      scriptrhmax=25
airmargin=28       airshift=yes
lowwater=20        highwater=25        purgelimit=5
timechar=:         outtime=30:00       readrate=180
localtimeout=20:00 remotetimeout=5:00  pausetimeout=20:00
logintimeout=1:00 netlogintimeout=3:00
min_passwd_length=5 maxhits=1000       timing_template=200
security=or        load=5
auto_upgrade=yes

```

## /site/wires/anpa7

```

;
; @(#) news/release/com/wires/anpa7 14.1.4.1
Date 00/08/06
; anpa7 - suitable for upi7, ap7, and cupi7
;
bits          7
start         <soh>
eos           <EOT>
category      -----
slug          30
;
;
map <16>      <SP>          ; TTS SPACE BAND
map <25>      <SP>          ; EM SPACE
map <29>      <SP>          ; THIN SPACE
map <30>      <SP>          ; EN SPACE
map <35>      .             ; EN LEADER
map <42>      ..            ; EM LEADER
map <60>      <23>          ; HARDEOL
map <61>      <23>          ; HARDEOL
map <62>      <23>          ; HARDEOL
map <64>      |N
map <91>      1/8
map <92>      1/4
map <93>      3/8
map <94>      |B
map <95>      --
map <123>     1/2
map <124>     5/8
map <125>     3/4
map <126>     7/8

```

## console.cfg

```
log b:log 32764
computer                ; A server
name                    a
label                   AVSTAR
irq                     3
hostess                 2c0
speed                   9600
;
computer                ; B server
name                    b
label                   AVSTAR
irq                     3
hostess                 2c8
speed                   9600
;
computer                ; C server
name                    c
label                   AVSTAR
irq                     3
hostess                 2d0
speed                   9600
;
modem                   ; remote console
password                issecret
timeout                 6:00
irq                     3
hostess                 2d8
speed                   2400
```

## SYSTEM.CLIENT.WINDOWS



*Either IP address or network card information is acceptable; IP address is preferable.*

```
125.1.100.1 ;02608cdbe7a2 ns001 big table
125.1.100.2 ;02608cd95e7e ns002 brock 07
125.1.100.3 ;02608c7e178e ns003 nydam 16
125.1.100.4 ;02608c7e67aa ns004 lockett 38
02608c7e519f ;ns005 michel 87
02608c7e1790 ;ns006 thibault 22
02608c7e51a8 ;ns007 ries 04
02608c7e6c01 ;ns008 lucas 57
125.1.100.9 ;02608c7e52c6 ns009 christensen 48
125.1.100.10 ;02608c7e5260 ns010 betty 69
125.1.100.11 ;0020aff431ff ns011 aveson 28
125.1.100.12 ;02608c7e6bfc ns012 robinson 49
125.1.100.13 ;02608c7e5274 ns013 tinsley 63
125.1.100.14 ;02608c7e6b58 ns014 reed 44
125.1.100.15 ;02608c7e532e ns015 landis 61
125.1.100.16 ;0020aff42efc ns016 dorr 51
125.1.100.17 ;0020aff42dad ns017 donnallen 60
125.1.100.18 ;0020aff42dcb ns018 douda 54
125.1.100.19 ;0020aff432ca ns019 kennedy 52
125.1.100.20 ;0020aff42d42 ns020 control room
125.1.100.21 ;0020aff431bf ns021 becker 30
125.1.100.22 ;0020aff43277 ns022 glass 29
```

## SYSTEM.CONFIGURE.301-ACTION

```
scan      system.shredder
remove

scan      system.cables.master
dup       system.cables.groups
dup       system.cables.cable#
dup       system.cables.device_type

scan      phones.airports
dup       phones.*all
scan      phones.business
dup       phones.*all
scan      phones.fire
dup       phones.*all
scan      phones.government.federal
dup       phones.*all
scan      phones.government.local
dup       phones.*all
scan      phones.government.state
dup       phones.*all
scan      phones.hospitals
dup       phones.*all
scan      phones.media
dup       phones.*all
scan      phones.misc
dup       phones.*all
```



## SYSTEM.MAP

```

;
;<run> <event dir> <composite><grp><mon off time>
;<name> <dev name> <update> <mct temp> <drv><dir><range>
;
show.6am.rundown events.6am - c 800
cg cg update - C:AMNEWS:500 599
cart cart - -

show.cutins.725 events.725 - c 900
cg cg update - C:AMNEWS:700 750
cart cart - -

show.noon.rundown-1 events.noon-1 - c 1400
cg cg update - C:AMNEWS:800 899
cart cart - -

show.noon.rundown-2 events.noon-2 - c 1400
cg cg update - C:AMNEWS:500 599
cart cart - -

show.6pm.rundown events.6pm - c 2100
cg cg update - C:PMNEWS:500 599
cart cart - -

show.11pm.rundown events.11pm - c 0300
cg cg update - C:PMNEWS:500 599
cart cart - -

show.updates events.updates - c 0100
cg cg update - C:PMNEWS:800 850
cart cart - -

show.special-3 events.specials - c 0200
cg cg update - C:PMNEWS:4000 4100
cart cart - -

show.sat-am.rundown events.sat-am - c 1100
cg cg update - C:PMNEWS:500 599
cart tape - -

show.sports-sat.rundown events.sports-sat - c 2200
cg cg update - C:PMNEWS:800 899
cart tape - -

show.sat-am.cut-in-1 events.sat-am-cut-in-1 - c 1100
cg cg update - C:AMNEWS:800 850
cart tape - -

```

```
show.sat-am.cut-in-2 events.sat-am-cut-in-2 - c 1100
cg cg update - C:AMNEWS:851 899
cart tape -

show.11pm.sportsroll events.sportsroll - c 0300
cg cg update - C:PMNEWS:1000 1040

show.6pm.sportsroll events.sportsroll - c 2100
cg cg update - C:PMNEWS:1050 1090
```

## SYSTEM.RESOURCE

;Dev	Style	Template	Effect	# lines	Comment
cg	bnews	62	-	0	;breaking news
cg	th	76	-	1	;today in history
cg	mm	75	-	1	;money matters
cg	pball	9996	-	6	;powerball
cg	lottopb	9993	-	12	;lotto/powerball
cg	cashpb	9994	-	9	;cash3/powerball
cg	world	50	-	1	;the world
cg	nation	50	-	1	;the nation
cg	fancash	9995	-	8	;fantasy 5 cash 3
cg	south	50	-	1	;the south
cg	recap	52	-	1	;recap
cg	nitetz	53	-	1	;11 tonight
cg	tvex	67	-	0	;tv exclusive
cg	uv	68	-	0	;unedited video
cg	tji	69	-	0	;this just in
cg	intvl	1	-	1	;1 line interview
cg	intv	2	-	2	;2 line interview
cg	intvdate	2	-	2	;1 line interview with date
cg	loc	4	-	1	;1 line locator
cg	locdate	5	-	2	;locator with date
cg	date	6	-	1	;date super
cg	file	7	-	0	;file tape
cg	newsfile	7	-	0	;newsfile
cg	ctsy	8	-	1	;courtesy
cg	sktch	10	-	1	;sketches by
cg	spkng1	11	-	1	;speaking one line
cg	spkng	12	-	2	;speaking two line
cg	scenes	13	-	1	;scenes from
cg	locskylv	24	-	1	;live skycam locator
cg	locsky	16	-	1	;locator skycam
cg	intvskylv	17	-	2	;interview skycam live
cg	repsky	18	-	1	;reporter skycam
cg	loclv	19	-	1	;live locator
cg	intvlv	20	-	2	;live interview
cg	intvllv	28	-	1	;live interview 1 line
cg	live	21	-	0	;live
cg	replv	22	-	1	;live generic reporter
cg	lvsky	23	-	0	;live skycam
cg	rep	27	-	1	;reporter not live
cg	sky	141	-	0	;skycam live
cg	frsky	254	-	0	;from skycam
;					
;special for NoonDay					
;					
cg	locn	1002	-	1	;1-line locator/noonday
cg	scenesn	1022	-	1	;scenes from/noonday

cg	innd	1001	-	2	;2 line interview noon
cg	lond	1008	-	1	;live locator noon
cg	inlvnd	1007	-	2	;live 2 line interview noon
;					
; SPORTS					
cg	nl	950	-	10	;national league
cg	al	951	-	10	;american league
cg	nfl	952	-	10	;nfl
cg	cf	953	-	10	;college football
cg	nba	954	-	10	;nba
cg	cbb	955	-	10	;college basketball
cg	nhl	956	-	10	;nhl
cg	prep	957	-	16	;high school
cg	scr	960	-	4	;score lower third
;; Anchors/Reporters					
cg	sa	104	-	0	; steve aveson
cg	salv	105	-	0	; steve aveson live
cg	sasky	107	-	0	; steve aveson skycam

## SYSTEM.WIRES.DISTRIBUTION

AP#medc2##	wires.medsource		
AP#env####	wires.environmental		
AP#####a#	wires.national		
AP#####c#	wires.features		
AP#####d#	wires.summaries		
AP#####e#	wires.features		
AP#####f#	wires.business		
AP#####g#	wires.state/regional		
AP#####h#	wires.summaries		
AP#####i#	wires.international		
AP#####j#	wires.state/regional		
AP#####m#	wires.farm		
AP#####n#	wires.state/regional		
AP#####o#	wires.weather		
AP#####p#	wires.politics		
AP#####q#	wires.sports.all		
AP#####q#	wires.sports.scores		
AP#####r#	wires.advisory.other		
AP#####s#	wires.sports.all		
AP#####s#	wires.sports.stories		
AP#nbc##t#	wires.newschannel		
AP#####t#	wires.advisory.other		
AP#####u#	wires.state/regional		
AP#####v#	wires.advisory.other		
AP#####v#	wires.daybook		
AP#####w#	wires.national		
AP#####	wires.unknown		
AP#####	wires.ap	ALWAYS	
AP#####	wires.all	ALWAYS	URGENT
NC#####u#	wires.advisory.priority		
NC#####	wires.all		
NC#####	wires.newschannel		
NC#####	wires.all	ALWAYS	URGENT
AP#####q#	sports.wires		
AP#####s#	sports.wires		

## SYSTEM.WIRES.KEYWORDS

* AP	
system.wires.keywords-ap	wires.keywords
system.wires.keywords-ap2	wires.
* UP	
*	
system.wires.keywords-others	wires.keywords

## SYSTEM.WIRES.KEYWORDS-AP

destination atlanta  
 atlanta  
 destination rape  
 rape  
 destination braves  
 braves  
 destination carter  
 carter  
 destination tuscaloosa  
 tuscaloosa and football  
 destination xgr  
 xgr  
 destination medsource  
 medc2 or medstar  
 destination gingrich  
 gingrich  
 destination space  
 (space AND shuttle) OR (shuttle AND columbia) or (shuttle  
 AND atlantis)  
 shuttle AND discovery  
 destination drugs  
 crack AND cocaine  
 destination air-safety  
 crash and plane OR ntsc or n.t.s.c. or faa or f.a.a.  
 destination delta  
 (delta and airlines) OR (hartsfield and airport)  
 destination guns  
 guns or weapons or nra

## SYSTEM.WIRES.KEYWORDS-AP2

destination 9am-specials  
 900a and specials





# APPENDIX C

## *Standard Dictionaries*

This appendix defines and explains contents of standard dictionaries as they are installed on your Avstar NRCS. Use this information for reference if modifying dictionary contents. Major sections are:

- Using Dictionaries to Define Messages and Commands
- Customizing Dictionaries
- Utility Messages Dictionary (/site/dict/messages)
- CCU Messages Dictionary (/site/dict/ccumsgs)
- Commands Dictionary (/site/dict/ccucmds)
- Video Attribute Dictionary (/site/dict/ccuvideo)
- Queues Dictionary (/site/dict/queues)
- Words Dictionary (/site/dict/words)
- Connect Dictionary (/site/dict/doac)
- Telex Dictionary (/site/dict/telex)
- Dial Dictionary (/site/dict/dial)
- Keyboard Macros Dictionary (/site/dict/keymacros)
- Printer Messages Dictionary (/site/dict/printmsg)
- Case-Shifting Dictionary (/site/dict/shift)
- VT Map Dictionary (/site/dict/vtmap)

## Using Dictionaries to Define Messages and Commands

Most commands, messages, and many queues your Avstar Newsroom Computer System (NRCS) uses are defined in dictionaries. Your system has a number of these dictionaries, each defining a particular group of commands, messages, or words, such as: the names of all commands are defined in the `ccucmds` dictionary. Many messages your system uses are defined in the `ccumsgs` dictionary.

Dictionaries let you customize your system's messages, workstation and commands, as well as the names of many of the queues your system uses (such as `SYSTEM.SEEK`). You can change the names of any of your system's required queues by editing their definitions in the `queues` dictionary and then having your system read the modified dictionary.



*If you customize your dictionary entries, keep a log of the changes you make so your changes can be re-entered after future software upgrades.*

Table C-1 lists names and locations of dictionary files.

**Table C-1 Dictionary Files**

File Name	Dictionary Name
/site/dict/ccucmds	Terminal Commands
/site/dict/ccumsgs	Error and Status Messages
/site/dict/ccuvideo	Video Modes
/site/dict/messages	Utility Messages
/site/dict/queues	Required Queues
/site/dict/words	Status and Option Words
/site/dict/doac	Connect Commands and Messages

**Table C-1 Dictionary Files (Continued)**

File Name	Dictionary Name
/site/dict/telex	Telex Commands and Messages
/site/dict/dial	Modem/Dial Messages
/site/dict/keymacros	Key Names for Macros
/site/dict/printmsgs	Printer Messages
/site/dict/shift	Case-Shifting Parameters
/site/dict/mcs	MCS Alarms and Words
/site/dict/mctcmds	MCT Commands
/site/dict/mctmsgs	MCT Error and Status Messages

Each line in a dictionary begins with a standard name followed by the translation for the command, message, or word represented by that standard name, as shown in Table C-2.

**Table C-2 Dictionary Standard Names**

Standard Name	Translation
cursor left	/cu l
cursor top	/cu t
cursor up	/cu u

Each translation is preceded by a slash (/). In commands, the translation represents the minimum a user must enter to execute the command. In the example above, the translation `cu u` is the minimum a user must enter to execute the `cursor up` command.

Your translations can be more than one word long, but your dictionaries have limited space, so keep each translation as short as possible. As

a rule, each translation should be just long enough to be unique from any other translation. For instance, the `cursor up` command is translated as `cu u`. This is the shortest translation for this command that keeps the translation unique.

If the purpose of a standard name or its translation is not clear, clarify them by inserting a comment on the line following the translation. Begin the comment line with a semicolon to prevent the system from treating it as part of the translation.



*Avstar NRCS uses standard names in each dictionary to match each translation to the correct command, message, or word. Do not change any of the standard names in your dictionaries.*

## Customizing Dictionaries

Your system's dictionaries are text files stored in the `/site/dict` directory. Because they are text files, you can change any dictionary translation using `ed`, the UNIX line editor at the console. See “`ed`, the Line Editor” on page 10-1 for more information.

## Changing Default Dictionary Values

As an example of how to modify a dictionary translation, change the `enter directory` and `enter queue` commands to make `direct` and `make queue`. To do this, change the translations for `enter directory` and `enter queue` from `e d` and `e q` to `ma d` and `ma q`, respectively.

The new translations, like the ones they are replacing, are as short as they can be and still remain unique. Keep each translation short, because there is a space limit.

If necessary, hide commands from users by changing the command name. For instance, to restrict access to the `order` command, change it

to something like `arrange` and inform only users that you want to have access to the command.

To edit the dictionary file, do the following:

1. Select *all* servers at the console, so changes you make are made to each server's copy of the file. See "Selecting One or More Servers" on page 2-5 for more information.

The `enter directory` and `enter queue` commands have their translations stored in `/site/dict/ccucmds`.

This procedure uses the UNIX line editor. See "Using the UNIX Line Editor" on page 10-4 for more information.

2. Open `/site/dict/ccucmds` for editing by typing:

```
ed /site/dict/ccucmds
```

A message similar to the following appears:

```
editing /site/dict/ccucmds
1824
```

3. Move to the line containing the standard name `enter directory` by typing:

```
enter directory /e d
```

4. Change `e d` to `ma d` by typing `s/e d/ma d`.

```
enter directory /ma d
```

5. Change the translation for `enter queue` in the same way.

```
enter queue /ma q
```

6. Type `w` (lowercase `w`) to save your changes to disk and type `q` to quit edit.

```
w
```

```
1826
```

```
q
```

To change a default dictionary value, do the following:

1. Edit the dictionary file.

See the previous procedure.

2. Type a message like the following:

```
Avstar-A: broadcast System going down at 1:55 pm. Please log out.
```

3. Type **offline** to bring the system offline. This will prevent users from logging in.
4. At the designated time, type **list t** to see whether anyone is still logged in.
5. A message similar to the following appears:

```
T13   carl   A
```

6. If anyone is logged in, type **logout all** to log them out.
7. Type **stop all** to stop *all* servers.
8. Become a console superuser. See “Becoming a Console Superuser” on page 3-2 for more information.
9. Type **makeccutab -b** (-b for build tables).

Use **makeccutab** to build the command and error message tables in the CCU program.

Information similar to the following appears:

```
Translating </site/dict/ccucmds>
Translating </site/dict/ccuvideo>
Translating </site/dict/ccumsgs> and </site/dict/queues>
<atccu> unused space is 205 characters
<directccu> unused space is 205 characters
```

After modifying a dictionary, run **makeccutab** and **maketab** to have your system read the modified dictionaries and incorporate changes into its programs.

The **makeccutab** command builds the command and error message table for the PCU dictionaries and then displays how much

space is unused in these dictionaries. In the previous example, the unused space is 205 characters.

10. Type **maketab -b** to build command and message tables and translate dictionaries for Avstar NRCS.

When the prompt returns, run **maketab** to translate site dictionaries. To make it build the table, add **-b** to this command.

A display similar to the following appears:

```
Translating </site/dict/queues>
Translating </site/dict/words>
Translating </site/dict/messages>
<host> unused space is 109 characters
Translating </site/dict/doac>
<snews> unused space is 221 characters
Translating </site/dict/doac>
<nxserver> unused space is 221 characters
```

The **maketab** command translates each dictionary and then displays unused space in these dictionaries.

11. If you changed the name of a command in a command dictionary, change the function key definition that references that command.
12. Restart *all* devices.
13. At the prompt, exit from superuser using CTRL-D.
14. Reset your system's PCUs/CCUs using their reset switches.
15. When all PCUs/CCUs have completed their reset, restart them by typing **restart all** at the Avstar console.

Once you have restarted the system, try using **enter directory** and **enter queue**. The system no longer recognizes these commands. It will recognize **make directory** and **make queue**.

16. Back up your site files with the **sitedump** command.

If you do not have one of your site files, a message similar to the following appears when you run the `makeccutab` or `maketab` console command:

```
Translating </site/dict/ccucmds>
Cannot access translation file </site/dict/ccucmds>.
Do you wish to use the standard English translations and
continue? (n/y)
```

If a set of dictionaries exceeds the amount of space allotted, a message similar to the following appears:

```
Table space exceeded by 14 characters
No modifications done!
```

## Restoring Dictionary Defaults

You can restore original translations without editing the dictionary again. Original dictionary files are held in the directory `/tmp/dict`, so even after you have made changes to a dictionary, you can restore standard translations by copying the appropriate dictionary file from `/tmp/dict` to `/site/dict` and running `makeccutab` and `maketab` again.

To restore standard translations for `enter` directory and `enter` queue, do the following:

1. List files in `/tmp/dict` by typing:

```
ls /tmp/dict
```

A display similar to the following appears:

```
ccucmds  ccuvideo  doac      messages  telex     utvmodes
ccmsgs   dial      queues   utcmds    words     keymacros
printmsg shift
```

2. Type `cp` followed by the pathname of the file you want to copy, and the pathname of the file you want to hold the copy.

```
cp /tmp/dict/ccucmds /site/dict/ccucmds
```



Once you have copied the file to `/site/dict`, complete the procedure for changing a translation.

## Utility Messages Dictionary (`/site/dict/messages`)

This dictionary holds a number of messages (see Tables C-3 through C-14) displayed in utility programs used by Avstar NRCS, including messages usually displayed when a user is building a form or creating a keyboard story. A few console messages are also included in this dictionary.

These messages do not contribute to the total size of translations, because they are sent to workstations only under special circumstances, and the system looks up translations only as they are needed rather than building them into a program.

**Table C-3 DBServer Program Messages**

Standard Name	Translation
M_NOSPACE	/NO SPACE IN SYSTEM
M_LOWSPACE	/SYSTEM LOW ON SPACE

**Table C-4 Server Program Messages**

Standard Name	Translation
M_PIOFAIL	/COMPUTER LINK FAILED
M_DISCONNECT	/COMPUTER DISCONNECTED

**Table C-5 Category and Keyword Check Program Msgs**

<b>Standard Name</b>	<b>Translation</b>
M_NODEST	/No destination found
M_DUPEDEST	/Duplicate destination
M_LINE	/Line
M_KWFMAX	/Too many keyword distribution files
M_BADQUEUE	/Could not enter the queue
M_NOTQUEUE	/Not a queue
M_PURGEZERO	/Queue is never purged
M_SYSERROR	/System error
M_CATLONG	/Category code word too long
M_CATMAX	/Too many category codes
M_CATFORM	/Illegal category format
M_CATHIDE	/Hidden category
M_KWDLONG	/Keyword too long
M_KWDMAX	/Too many keywords
M_WNOTLAST	/Default keyword list must be last
M_SYNEROR	/Syntax error
M_MISSING	/Missing
M_UNEXPCTD	/Unexpected
M_FILENUM	/Maximum file number bad

**Table C-6 Keyboard Check Program Messages**

Standard Name	Translation
M_COMPUTER	/Computer
M_KEYDUP	/Duplicate key description
M_KEYFUNKY	/Warning: badly placed @ exists in key definition line
M_KEYRANGE	/Invalid key number
M_KEYREP	/Warning: a key definition contains a repeating function
M_KEYSEP	/Missing key number separator (~)
M_KEYSTART	/First key description does not begin with @
M_KEYMIN	/Not enough key descriptions
M_KEYLONG	/Keyboard description contains too many characters
M_KEYOK	/Keyboard ok
M_KEYBAD	/Keyboard NOT usable

**Table C-7 Keyboard Check Messages for Macros**

Standard Name	Translation
M_MACRO	/%s macro #%d:
M_NOLOCATE	/Could not locate "%c%d"
M_BADMEMORY	/Memory allocation error
M_REFERENCE	/Circular reference to macro #%d:
M_BADSTACK	/Unable to stack keywords

**Table C-7 Keyboard Check Messages for Macros (Continued)**

<b>Standard Name</b>	<b>Translation</b>
M_MISMATCH	/Mismatched "%c%c"
M_TWOTILDES	/Multiple "%c"s found
M_RESWORD	/No "%c%c" found for reserved word %s
M_TWOTAGS	/Multiple macro keys: %s %s
M_NOTILDE	/No "%c" found
M_NOTAG	/No macro key tag
M_TWODEFS	/Duplicate macro definition:
M_LONESTATE	/Isolated keyboard state:
M_DISTRIBUTE	/%s does not distribute
M_EMPTY	/Empty macro
M_STDHELP	/Warning: "Help" key redefined:
M_STDFINDNEXT	/Warning: "Find Next" key redefined:
M_STDEXIT	/Warning: "Avstar Exit" key redefined:
M_STDCLOSE	/Warning: "Window Close" key redefined:
M_STDDISCARD	/Warning: "Discard Changes" key redefined:
M_STDREFRESH	/Warning: "Refresh" key redefined:
M_STDSSCRIPT	/Warning: "Script Swap" key redefined:
M_STDWINNEXT	/Warning: "Next Window" key redefined:

**Table C-8 Wire Program Messages**

Standard Name	Translation
M_WIREFAIL	/HOST-CCU COMMUNICATIONS ERROR
M_WIREIDLE	/Wire has been idle for (time indicated)
M_WIRERESUME	/Wire received story, was idle for (time indicated)

**Table C-9 Mail Server Messages**

Standard Name	Translation
M_MAILSYNTAX	/Cannot send mail, no address
M_MAILNOREC	/Unable to receive mail from name
M_MAILQUEUE	/Cannot return mail, bad mail queue

**Table C-10 Validation (Action) Server Messages**

Standard Name	Translation
M_VALID	/Story valid
M_INVALID	/Story invalid
M_VMOVEDTO	/Story invalid-Moved to

**Table C-11 Seek Server Messages**

<b>Standard Name</b>	<b>Translation</b>
M_BGSNSCHP	/No search path
M_BGSNRESP	/No results path
M_BGSKWLNG	/Keyword expression too long
M_BGSSCHTP	/Invalid Search Type
M_BGSFLDMS	/Field missing from form
M_BGSIKWEX	/Invalid keyword expression
M_BGSRQDNE	/Results Queue does not exist
M_BGSRQNAQ	/Results Queue not a queue
M_BGSRQNPM	/No write permission for results queue
M_BGSINVLD	/Invalid Search
M_BGSEOP	/End of path
M_BGSMAXH	/Max hits
M_BGSSPDNE	/Search Path does not exist
M_BGSSPI	/Invalid search path
M_BGSRQI	/Invalid results queue
M_BGSRQOE	/Open error on results queue
M_BGSACCESS	/Access denied

**Table C-12 Last Login Messages**

Standard Name	Translation
M_LASTLOG	/Last login
M_ONDEVICE	/On device

**Table C-13 Messages for the Print Server**

Standard Name	Translation
M_PRINTER_BUSY	/Printer is OFFLINE

**Table C-14 Messages for the Sony Barcode Printer**

Standard Name	Translation
M_NOBULKPR	/Bulk printing disabled

## CCU Messages Dictionary (/site/dict/ccumsgs)

Table C-15 shows entries in the CCU messages dictionary, with their default translations.

**Table C-15 CCU Messages Dictionary**

Standard Name	Translation
D_AIRBEND	/End-backwards
D_AIRFEND	/End-forward
D_AIRSTOP	/Stopped

**Table C-15 CCU Messages Dictionary (Continued)**

<b>Standard Name</b>	<b>Translation</b>
D_AUTHORIZED	/Not allowed
D_BACKT	/Backtime
D_BADARG	/Bad argument
D_BADDEST	/Bad destination
D_BADKBD	/Bad keyboard
D_BADREP	/Must FIND First
D_BUSY	/Busy
D_CANTFIND	/Can't find
D_CMD	/CMD
D_CONTINUE	/Press CMD to continue
D_DEFINED	/Text is defined
D_DIRMODE	/Not in directory
D_DUPED	/2 spaces then DUP'D
D_EMPTY	/5 spaces then EMPTY
D_ERROR	/System error
D_EXISTS	/Already exists
D_GONE	/Gone
D_HOME	/home
D_INSERT	/Insert
D_KEY	/Key (must match ccucmd translation)
D_KILLED	/1 space then KILLED



**Table C-15 CCU Messages Dictionary (Continued)**

<b>Standard Name</b>	<b>Translation</b>
D_LINE	/Line
D_LOCKED	/Locked
D_LOGGEDIN	/Logged in
D_LOGIN	/Login (must match ccucmd translation)
D_MAIL	/mail
D_MAILSTORY	/Story
D_MAXSONS	/Too many
D_MESSAGE	/Message
D_MODE	/Invalid in this mode
D_MOVED	/2 spaces then MOVED
D_NAMETL	/Name too long
D_NEW_PASSWORD	/New password
D_NOARG	/Needs argument
D_NOCMD	/Unrecognized command
D_NODEFINED	/Nothing defined
D_NODELETED	/Nothing deleted
D_NOFINDSTR	/Type word to find
D_NOLOG	/Not logged in
D_NOMSG	/No messages
D_NOREPSTR	/Type new word
D_NOTBATCH	/Queue lacks that capability

**Table C-15 CCU Messages Dictionary (Continued)**

<b>Standard Name</b>	<b>Translation</b>
D_NOTEMPTY	/Not empty
D_NOTEXIST	/Story moved
D_NOUSER	/No such user
D_OFFLINE	/Offline
D_ONAIR	/On air
D_ORDER	/Order
D_PASSWORD	/Password (must match ccucmd translation)
D_PAUSED	/Paused
D_PERM	/P
D_PRINTED	/PRINTED
D_PURGELOCK	/Queue busy
D_QONLY	/Only in queue
D_QUEMODE	/Not in queue
D_REFRESH	/Refresh
D_REMERROR	/Remote link error (only for split remote)
D_REMOTE	/Remote (only for split remote)
D_REPLACE	/Replace
D_RDONLY	/Read-only
D_SORRY	/Sorry
D_SORTED	/Sorted
D_STORY	/Story

**Table C-15 CCU Messages Dictionary (Continued)**

Standard Name	Translation
D_SYNTAX	/Bad usage
D_TOO_SHORT	/Too short
D_TXED	/TRANSFERRED
D_UNABLEMAIL	/Unable to send mail
D_UNKNOWN	/Unknown command
D_WAIT	/Please stand by
D_WORD	/Word

## Commands Dictionary (/site/dict/ccucmds)

The commands dictionary (Tables C-16 and C-17) contains translations for your system's commands. Your system's version of this file may be slightly different.

You can customize commands entered at the VT terminal by changing the translation definition. For instance, the default translation of `backtime next` is `ba n`, as shown in Table C-16. You can change the translation to `back n` or `back next`, or some other translation.

## Console Commands

Table C-16 lists console commands in the Commands Dictionary.

**Table C-16 Console Commands (/site/dict/ccucmds)**

Standard Name	Translation
air	/ai
air prompter	/ai p
air scrollbox	/ai s
backtime	/ba
backtime begin	/ba b
backtime end	/ba e
backtime next	/ba n
backtime pause	/ba p
blank	/bl
bulletin	/bu
call	/ca
connect	/co
cursor bottom	/cu b
cursor down	/cu d
cursor left	/cu l
cursor right	/cu r
cursor top	/cu t
cursor up	/cu u
define character	/def c

---

**Table C-16 Console Commands (/site/dict/ccucmds) (Continued)**

<b>Standard Name</b>	<b>Translation</b>
define erase	/def e
define paragraph	/def p
define word	/def w
delete character	/del c
delete defined	/del d
delete recover	/del r
destination	/des
display	/di
display no refresh	/di n
display refresh	/di r
display story	/di s
duplicate	/du
enter directory	/e d
enter queue	/e q
find	/fi
flash	/fl
get	/ge
get all	/ge a
get old	/ge o
go	/go
help	/he

**Table C-16 Console Commands (/site/dict/ccucmds) (Continued)**

<b>Standard Name</b>	<b>Translation</b>
hold	/ho
insert block	/i b
insert character	/i c
insert mode	/i m
key	/ke
kill	/kill
load keyboard	/loa k
lock	/loc
login	/logi
logout	/logo
logout exit	/logo e
mail	/ma
mail copy	/ma c
mail reply	/ma r
message	/me
message clear	/me c
message recall	/me r
monitor	/mon
move	/mov
new	/ne
new bottom	/ne b

**Table C-16 Console Commands (/site/dict/ccucmds) (Continued)**

<b>Standard Name</b>	<b>Translation</b>
new insert	/ne i
new top	/ne t
note clear	/no c
note recall	/no r
note save	/no s
order	/or
pack	/pac
pack all	/pac a
page down	/pag d
page up	/pag u
password	/pas
pause	/pau
print	/pr
quit	/q
read	/rea
read all	/rea a
read old	/rea o
receive	/receive
remove	/rem
replace	/rep
save	/sa

**Table C-16 Console Commands (/site/dict/ccucmds) (Continued)**

<b>Standard Name</b>	<b>Translation</b>
script	/scri
script undo	/scri u
scroll bottom	/scro b
scroll down	/scro d
scroll top	/scro t
scroll up	/scro u
seek	/see
send	/sen
split	/sp
status clear	/st c
tab right	/ta r
form recall	/fo r
form save	/fo s
time stamp	/ti s
transmit	/tr
unlock	/unl
unsplit	/uns
user	/us
video define	/vid d
video step	/vid s
view	/vie

---



---

## Job List Commands

Table C-17 lists Job List Commands in the Commands Dictionary.

**Table C-17 Job List Commands (/site/dict/ccucmds)**

<b>Standard name</b>	<b>Translation</b>
scan	/scan
bscan	/bscan
open	/open
put	/put
ignore_del	/ignore-del
send_del	/send-del
eof	/eof
on	/on
at	/at
every	/every
validate	/validate
priority	/priority
ignore	/ignore
quiet	/quiet
verify	/verify
order	/order
fullform	/fullform
number	/number

---

## Video Attribute Dictionary (/site/dict/ccuvideo)

This dictionary (see Table C-18) controls names of video attributes and the order in which they appear when you press the video key or enter the `video step` command. It also controls which attributes are available. Like the CCU dictionary, standard names in this dictionary are in uppercase while translations can be in uppercase, lowercase, or mixed case.

Unlike other dictionaries, you can delete entries and change the order of entries in this dictionary. Changing the order in which video attributes are listed changes the order in which they appear when you press the video key or enter the `video step` command.

If you do not use all video attributes at your station, delete standard names for those you do not use. This saves some space and enables you to avoid having to step through unused video attributes to get to one you want to use. This is the only time you can remove a standard name from a dictionary file.

Typical video attributes and translations are listed in Table C-18. The absence of a translation for `NORMAL` means that nothing is displayed when you reach this part of the cycle, but the `NORMAL` video attribute still functions.

**Table C-18** Video Attribute Dictionary

Standard Name	Translation
NORMAL	/
BOLD	/newscaster
REVERSE	/video
UNDERLINE	/underlined
BOLDREVERSE	/director
\$\$	The \$\$ indicates the end of the video step list. Any lines after that are ignored.

## Queues Dictionary (/site/dict/queues)

The queues dictionary (see Table C-19) contains names for system queues such as `SYSTEM.KEYBOARDS` and the Dead queue.

Information on the queues dictionary as defined for the FTS server is in Table 14-2 on page 14-70.

Queues in this dictionary are used by functions within Avstar NRCS. For instance, the `seek` command uses whatever queue translation is given to `Q.SEEK`, which is `SYSTEM.SEEK` by default. Like other dictionaries, the standard name is in uppercase and must not be changed. The translation can be in lowercase, but appears in uppercase on the screen. Queue names and their standard translations are shown in Table C-19. The 8-bit codes can be defined using 7-bit sequences.

**Table C-19 Queues Dictionary**

Standard Name	Translation
Q_ACCT	/system.account
Q_CLIENT_DOS	/system.client.dos
Q_CLIENT_WINDOWS	/system.client.windows
Q_CLIENT_VERSIONS	/system.client.versions
Q_FORMS	/system.forms
Q_KEYBOARDS	/system.keyboards
Q_MESSAGE	/system.message
Q_CATWORDS	/system.wires.distribution
Q_KEYWORDS	/system.wires.keywords
Q_SEARCHTAPE	/system.searchtape
Q_CONFIGURE	/system.configure
Q_UNKNOWN	/wires.unknown
Q_DEAD	/dead

**Table C-19 Queues Dictionary (Continued)**

<b>Standard Name</b>	<b>Translation</b>
Q_NODEST	/system.unknown
Q_FLASH	/wires.advisory.priority
Q_STYLES	/system.styles
Q_PRINTERS	/system.printers
Q_ADDRLIST	/system.teleprinter.addrlist
Q_ADDRDIST	/system.teleprinter.distribution
Q_HELP	/system.help.terminal
Q_SERVICE	/system.service
Q_SCRIPT	/system.dialogs
Q_MRESOURCE	/system.resource
Q_MMAP	/system.map
Q_MAILOUT	/system.mail.out
Q_MAILERROR	/system.mail.error
Q_SEEK	/system.seek
Q_GROUPS	/system.groups
Q_PRINTER2	/system.printers2
Q_SPELL	/system.spell
Q_USERROOT	/people
Q_HOME	/
Q_DESTINATION	/notes
Q_MAIL	/mail

## Words Dictionary (/site/dict/words)

The words dictionary (see Tables C-20 through C-23) contains translations for a variety of miscellaneous words used by the system. For instance, words regarding priority (such as flash and silent) or print options (such as story and script) are included.

Information on the words dictionary as defined for the FTS server is in Table 14-2 on page 14-70.

Standard names are in uppercase and must not be changed. Because many messages in this dictionary are displayed in the upper right corner of active stories and rundowns, keep them short to avoid overwriting portions of the story or rundown. Translations can be uppercase, lowercase, or mixed case.

**Table C-20 Wire Priorities and Options**

Standard Name	Translation
W_FLASH	/FLASH
W_BULLETIN	/BULLETIN
W_URGENT	/URGENT
W_SILENT	/SILENT
W_ALWAYS	/A
W_TRANSMIT	/TRANSMIT
W_MAIL	/mail

**Table C-21    Status Types**

<b>Standard Name</b>	<b>Translation</b>
W_HOLD	/HOLD
W_LOCKED	/LOCKED
W_READY	/READY
W_NEW	/NEW
W_WIRE	/WIRE

---

**Table C-22    Special Words for Find**

<b>Standard Name</b>	<b>Translation</b>
W_AND	/and
W_NOT	/not
W_OR	/or
W_FINDTIME	/time
W_DEFAULT_FORM	/default_form
W_WEBACC_FORM	/access_form
W_WEBPUB_FORM	/publish_form
W_SLOW	/slow
W_ALL	/all

---

**Table C-23 Print Command Options**

<b>Standard Name</b>	<b>Translation</b>
W_STORY	/story
W_SCRIPT	/script
W_RUNDOWN	/rundown
W_DIRECTORY	/directory
W_ON	/on
W_COPIES	/copies
W_STYLES	/style
W_LINES	/lines
W_OUTTIME	/outtime
W_SPACING	/spacing
W_PRINT	/print
W_DAYS	/SunMonTueWedThuFriSat  You can change these translations only once. If you make a mistake, or want to change them again, you must extract the news program from the release CD first. Call iNews Customer Support for assistance in extracting the program.
W_MONTHS	/JanFebMarAprMayJunJulAugSepOct NovDec  You can change these translations only once. If you make a mistake, or want to change them again, you must extract the news program from the release CD first. Call iNews Customer Support for assistance in extracting the program.
W_PAGE	/page

**Table C-23 Print Command Options (Continued)**

<b>Standard Name</b>	<b>Translation</b>
W_LOGTYPES	/DRU
W_ERROR	/ERROR (only seek)
W_INDXD	/archvd
W_DEST	/destination
W_QUEUE	/queue
W_START	/ON
W_OFF	/OFF
W_LOAD	/LOAD
W_UNLOAD	/UNLOAD
W_YES	/yes
W_NO	/no
W_GROUP	/group
W_ALIAS	/alias
W_ANYSTR	/-
W_BLANKSTR	/+

**Table C-24 Words Relating to the Seek Server**

<b>Standard Name</b>	<b>Translation</b>
W_FAST	/fast
W_ACTIVE	/ACTIVE
W_DONE	/DONE



**Table C-24 Words Relating to the Seek Server (Continued)**

Standard Name	Translation
W_ABORT	/ABORTED
W_RESTRICTED	/restricted (only gtraits)
W_MBSPACE	/*À  This token applies to multibyte Avstar systems only. It must be set to the multibyte space character of the code set being used.
W_PAGEBREAK	/pagebreak
W_DELIMITERS	/ <code>&lt;sp&gt; “()-.’/</code>
W_UMANAGER	/UMANAGER
W_PENDING	/PENDING
W_BREAK	/= = =

## Connect Dictionary (/site/dict/doac)

The connect dictionary (see Table C-25) contains messages and commands used by the Connect feature.

The standard name is in uppercase and must not be changed. The translation can be in lowercase, uppercase, or mixed case.

**Table C-25 Connect Dictionary**

Standard Name	Translation
S_SNPROMPT	/cmd>
S_SNCAPON	/Capturing session to

**Table C-25 Connect Dictionary (Continued)**

<b>Standard Name</b>	<b>Translation</b>
S_SNCAPOFF	/Session saved to
S_SNPAUSE	/Pause capture
S_SNESCAPE	/New escape character
S_SNQUIT	/Quitting
S_SNCAPERR	/Capture error!
S_SNNOQUEUE	/Could not append to queue
S_SNCRERR	/Error creating capture story
S_SNCLOSED	/Connection closed
S_NOSVC	/Unknown service
S_NOCAPTURE	/Session not saved to
S_SNNOTCAP	/Not capturing
S_SNEXPECT	/Failed to get expected string
S_NSCRIPT	/Could not open script story
S_CONNECT	/CONNECT
S_ACCEPT	/ACCEPT
S_REJECT	/REJECT
S_FINISH	/FINISH
S_TERMINAL	/TERMINAL
S_RXNET	/RXNET
S_RXDNET	/RXDNET
S_CCAPTURE	/capture

**Table C-25 Connect Dictionary (Continued)**

<b>Standard Name</b>	<b>Translation</b>
S_CQUIT	/quit
S_CPAUSE	/pause
S_CSTOP	/stop
S_CHELP	/help
S_CESCAPE	/escape
S_CECHO	/echo
S_CHEOL	/heol
S_CTYPE	/type
S_CWAIT	/wait
S_CMESS	/message
S_CEXPECT	/expect
S_CDELAY	/delay
S_CTIMER	/timer
S_CPASS	/pass
S_CDIAG	/diag
S_CMAP	/map
S_CMAPIN	/mapin
S_CMAPOUT	/mapout
S_CAPHLP	/<queue> Capture to queue, or continue after a pause
S_QUITHLP	/End this connect session

**Table C-25 Connect Dictionary (Continued)**

Standard Name	Translation
S_PAUSEHLP	/Pause capture, but do not close capture story
S_STOPHLP	/Stop capturing and close capture story
S_HELPHLP	/Show this list of commands
S_ESCHLP	/ <i>&lt;c&gt;</i> Change escape character to specified character
S_HEOLHLP	/Toggle Hard-End-Of-Line on captured data
S_ECHOHELP	/Toggle local character echo
S_MAPHLP	/ <i>&lt;fromchar&gt;</i> <i>&lt;tochar&gt;</i> map input & output
S_MAPINHLP	/ <i>&lt;fromchar&gt;</i> <i>&lt;tochar&gt;</i> map input
S_MAPOUTHLP	/ <i>&lt;fromchar&gt;</i> <i>&lt;tochar&gt;</i> map output
S_NXNONAME	/Computer not named
S_NXNOCONF	/System not configured
S_NXNOPTY	/No ptys available
S_NXNODEV	/No device available
S_KTRANSFER	/transfer
S_KCANCELLED	/cancelled
S_KDISCARDED	/discarded
S_KLOCAL	/L:
S_KREMOTE	/R:
S_KEOF	/[EOF]

**Table C-25 Connect Dictionary (Continued)**

Standard Name	Translation
S_KEOT	/[EOT]
S_KTIMEOUT	/Timed out
S_KUNKNOWN	/Unknown packet type
S_KIOPENERR	/Input file open error
S_KINOTFOUND	/File not found
S_KOOPENERR	/Output file open error
S_KOWRITERR	/Output file write error
S_KOCLOSERR	/Output file close error

## Telex Dictionary (/site/dict/telex)

The telex dictionary (see Table C-26) contains commands and messages used with the telex feature.

**Table C-26 Telex Dictionary**

Standard Name	Translation
X_0	/LINE FREE
X_1	/LINE UNAVAILABLE
X_10	/RING INDICATE
X_20	/CONNECTION MADE
X_36	/INCORRECT AB
X_43	/NO CONNECTION

**Table C-26 Telex Dictionary (Continued)**

<b>Standard Name</b>	<b>Translation</b>
X_50	/CALL CLEARED
X_61	/INVALID COMMAND
X_70	/SS ABS
X_71	/SS DER
X_72	/SS MON
X_73	/SS NCH
X_74	/SS NP
X_75	/SS NA
X_76	/SS OCC
X_77	/SS NC
X_81	/SS INF
X_82	/SS CIC
X_83	/SS RDI
X_DISTRIBUTE	/SYSTEM.TELEX.DISTRIBUTION
X_UNKNOWN	/dead
X_CMD	/CMD
X_SENT	/SENT
X_ABORT	/ABORT
X_BUSY	/BUSY
X_NODIAL	/NODIAL
X_NOCNX	/NOCNX

**Table C-26 Telex Dictionary (Continued)**

Standard Name	Translation
X_MORE	/MORE
X_RETRY	/RETRY
X_FAIL	/FAIL
X_MLKUP	/NO LOOKUP ENTRY FOUND
X_MMORE	/SENT OK REMAINING SENDS:
X_MSENT	/SENT OK
X_MRETRY	/NO CONNECTION-RETRIES LEFT:
X_MFAIL	/FAILED-NO MORE RETRIES

## Dial Dictionary (/site/dict/dial)

The dial dictionary (see Table C-27) contains messages used with modems.

Like other dictionaries, the standard name is in uppercase and must not be changed. The translation can be in lowercase, uppercase, or mixed case.

**Table C-27 Dial Dictionary**

Standard Name	Translation
C_USAGE	/Dial: bad usage
C_MDMTO	/Modem timed out
C_NOCONS	/Could not open /dev/console
C_MDMRDY	/Line ready..

**Table C-27 Dial Dictionary (Continued)**

Standard Name	Translation
C_DSYSERR	/Dial: system error
C_MDMNOT	/Modem not responding
C_MDMNODIAL	/Modem not responding to dial command
C_MDMNOHUP	/Modem did not hang up—Check modem
C_MDMHUP	/Modem is hung up
C_QUITQ	/Do you really want to quit (y/n)?

## Keyboard Macros Dictionary (/site/dict/keymacros)

The keyboard macros dictionary (shown in Table C-28) contains names of keyboard keys for use in keyboard macro definitions for Avstar NRCS.

Like other dictionaries, the standard name is in uppercase and must not be changed. Translations can be in lowercase, uppercase, or mixed case.

**Table C-28 Keyboard Macros Dictionary**

Standard Name	Translation
K_NULL	/null
K_F1	/f1
K_F2	/f2
K_F3	/f3
K_F4	/f4
K_F5	/f5



**Table C-28 Keyboard Macros Dictionary (Continued)**

<b>Standard Name</b>	<b>Translation</b>
K_F6	/f6
K_F7	/f7
K_F8	/f8
K_F9	/f9
K_F10	/f10
K_F11	/f11
K_F12	/f12
K_KP0	/kp0
K_KP1	/kp1
K_KP2	/kp2
K_KP3	/kp3
K_KP4	/kp4
K_KP5	/kp5
K_KP6	/kp6
K_KP7	/kp7
K_KP8	/kp8
K_KP9	/kp9
K_INSERT	/insert
K_HOME	/home
K_PAGEUP	/pageup
K_PAGEDOWN	/pagedown

**Table C-28 Keyboard Macros Dictionary (Continued)**

<b>Standard Name</b>	<b>Translation</b>
K_DELETE	/delete
K_END	/end
K_UP	/up
K_DOWN	/down
K_LEFT	/left
K_RIGHT	/right
K_SHIFT	/shift
K_CTRL	/ctrl
K_ALT	/alt
K_TAB	/tab
K_ESC	/esc
K_BACKSPACE	/backspace
K_ENTER	/enter
K_PAUSE	/pause
K_REPEAT	/repeat

---

## Printer Messages Dictionary (/site/dict/printmsgs)

The printer messages dictionary (shown in Table C-29) contains error messages generated by the Sony barcode printer.

The standard name is in uppercase and must not be changed. The translation can be in lowercase, uppercase, or mixed case.

**Table C-29 Printer Messages Dictionary**

Standard Name	Translation
M_TAPEIDREQ	/Tape ID required
M_INVTAPEID	/Invalid tape ID
M_DUREXCEED	/Duration time exceeded
M_DURTOOSHORT	/Duration time too short
M_INVTIMECODE	/Invalid timecode
M_EOMREQ	/EOM required
M_SOMREQ	/SOM required
M_INVCHARS	/Tape ID or title has invalid characters
M_EOMSTMISSING	/EOM Style missing from profile
M_TAPEIDLEN	/Tape ID must be between 3-8 characters long
M_PAGESTRLEN	/Page string too long
M_SOMEXCEEDS	/SOM timecode exceeds EOM timecode
M_UNKERR	/Unknown print error
M_CMDDEL	/*

## Case-Shifting Dictionary (/site/dict/shift)

The case-shifting dictionary maps lowercase characters to their uppercase counterparts and vice versa. Avstar NRCS shifts the case of a character according to its decimal value in a standard character conversion table.

The dictionary has two parts:

- The first part, labeled with the keyword `tolower`, maps decimal values of uppercase characters to the decimal values of their lowercase counterparts.
- The second part, labeled with the keyword `toupper`, maps decimal values of lowercase characters to decimal values of their uppercase counterparts.

In the default dictionary shipped with Avstar NRCS, a character at a decimal position in the range on the left of the arrow (->) shifts to the character at the corresponding decimal position in the range on the right. For instance, the character at decimal position 65 (A) is mapped to the character at decimal position 97 (a); the character at decimal position 66 (B) shifts to the character at decimal position 98 (b); and so on:

```
;
tolower
 65 - 90 -> 97 - 122      ; A - Z -> a - z
192 - 207 -> 224 - 239
209 - 221 -> 241 - 253
end

toupper
 97 - 122 -> 65 - 90      ; a - z -> A - Z
224 - 239 -> 192 - 207
241 - 253 -> 209 - 221
end
```

The character-conversion table the system uses depends on the interface you are using:

- If you are using the Avstar DOS or VT100 interfaces, the character-conversion table is based on the DEC Multinational Character Set (MCS).
- If you are using the Avstar Graphic User Interface (GUI), the conversion table is based on the ISO standard for multinational characters.

If character mappings specified in these standard character-conversion tables are not appropriate for the language you are using, edit the `/site/dict/shift` file to remap character conversions. You can map ranges of values (as shown in the default dictionary file) or you can map values one by one, if necessary.

As you edit the dictionary file, follow these guidelines:

- Ensure all keywords (`tolower`, `toupper`, `end`) in the dictionary file remain in lowercase.
- Specify all character-conversions in terms of the characters' decimal values in the conversion table. Do not specify a value higher than 255.
- Any characters not mapped in the dictionary file remain the same when shifted.
- The system ignores blank lines in the dictionary file and any characters following a semicolon (;).

After you edit the dictionary file, run the `makeshift` console command in maintenance mode during installation to prepare the case-shifting dictionary to be used by Avstar NRCS. For more information, call iNews Customer Support.

If you map a character to more than one value, the system displays a warning when you type the `makeshift` command, but uses the last character mapping in the file.

## VT Map Dictionary (/site/dict/vtmap)

The `makeccutab` looks for an optional dictionary file named `/site/dict/vtmap`. If present, it will be used to construct a map to convert character input from the Video Terminal (VT) and stored into the Avstar database and vice versa—to convert characters stored in the database to original character input from the VT. This is symmetrical mapping.

The format of lines in `/site/dict/vtmap` is:

```
map      <vt code>      <database code>
```

The `<vt code>` can only be in the range 144 to 255. The `<database code>` can be in the range of 128 to 255.



*The `<vt code>` values of 128 to 143 are reserved for system keys. There is no restriction on the number of maps. An identity map is assumed for all `<vt code>` values not explicitly mapped—that is, `<vt code>=<database code>`. The map table requires 256 bytes of string space in the CCU program.*

Only the CCU program will use this map table. The VT sessions from AnDOS and network terminals will not use this map table. Connect session started from a VT session on a CCU will be subject to this defined mapping.

The minimum set of programs required to implement this functionality is:

```
/exc/makeccutab  
/exc/ccu/at/ccu  
/exc/ccy/direct/ccu  
/exc/ccu/tftp/ns/ns.exe  
/exc/news
```

# APPENDIX D

## *PCU/CCU Reference*

This Appendix contains the following major sections:

- Overview
- Network CCUs
- CCU II
- CCU III
- CCU IV
- PCU

## Overview

A Peripheral Controller Unit (PCU) is a Personal Computer (PC) connected between an Avstar Server and one or more serial devices, such as printers and wire services. The PCU expands the available serial ports and relieves the server of routine communication with the devices. The PCU is an open system. The operating system is Microsoft DOS and a standard public domain TCP/IP network stack is used.

A Communications Concentrator Unit (CCU) is a different model of the same type of device. The CCU operating system and network stack is proprietary to Avstar.

No server should be connected to more than half the number of PCUs or CCUs it is capable of supporting. This ensures that if one server fails, the surviving server(s) have enough reserve capacity to run the failed server's PCUs or CCUs.

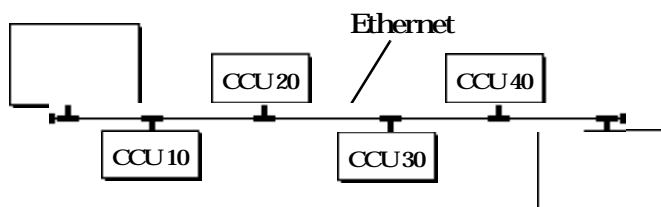
Divide your system's PCUs or CCUs evenly among your system's servers. This yields maximum performance possible from each server. If a server fails, reconfigure the failed server's PCUs or CCUs to the surviving servers until the failed server is repaired.

Your system may use four types of PCUs or CCUs: CCU II, CCU III, CCU IV, and PCU. While these are similar in operation and appearance, there are a few differences. This appendix discusses what is common to each type of PCU or CCU, as well as the features that are unique to each type.



## Network CCUs

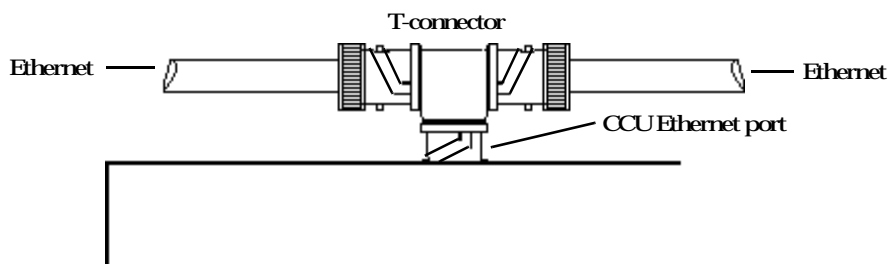
As illustrated in Figure D-1, network CCUs connect to the Ethernet network. Your system's servers also connect to this network and use it to communicate with their CCUs.



**Figure D-1 Servers and Network CCUs**

Network CCUs are delivered with an Ethernet card installed and with the internal switches set to enable network communication.

Each network CCU uses a T-connector to connect to the network. The T-connector's stem attaches to the CCU Ethernet port, and its arms attach to the network cables. Figure D-2 is 10-base2 thin=net.



**Figure D-2 CCU-Ethernet Connection**



**To disconnect a CCU from the network, stop the CCU and then disconnect it from the T-connector. Ensure that you disconnect the T-piece from the CCU Ethernet card; do not disconnect the Ethernet connections to the T-piece as this will cause your entire network to**

**fail. Do not disconnect the T-connector from the network unless you have shut down the system—otherwise, the network connection fails and you may lose data.**

## Connecting Devices to a CCU

Serial devices, such as wires and printers, connect to CCUs through serial ports on the CCU back panel. The devices you connect to a CCU must be capable of standard RS-232 serial communication.

If you want to connect a non-RS-232 device, such as a low-speed wire or a Sony betacart (RS-422), to a CCU, use an interface adapter that converts the device's output to a standard serial transmission and the CCU output from standard serial transmission to a format the device uses. If you are unsure how to do this, call iNews Customer Support for assistance.



*When connecting a device to a CCU, connect the serial cable to the device first and then connect the other end of the cable to the CCU. If you connect the cable to the CCU first, it acts as an antenna, picking up static and noise until you connect the other end. This static and noise may confuse the CCU to a point where the port stops working. If this happens, disconnect the cable and restart the CCU.*

## Physical Specifications for CCU II and CCU III

Your CCUs must meet the following physical specifications/dimensions.

**Table D-1 CCU/PCU Dimensions**

Specification	Minimum Value
Power rating	110 volts or 220 volts, 50-60 Hz
Power consumption	70 watts
CPU type	Intel 80286, 80386, or 80486
CPU speed	10 MHz or better
Memory	1 megabyte
Heat	220 BTU/hr.
Dimensions	7" H x 19" W x 21 3/4" D CCUs are installed in a standard 19" equipment rack, and each is four rack units high (7").
Weight	26 lbs.
Shipping weight	35 lbs.

These specifications do not apply to PC-type PCUs.

### Environmental Requirements

While your CCUs are very stable devices and can operate in a wide range of temperatures and humidity, your system performs better if you maintain optimal operating conditions.

Maintain the server room at 72° Fahrenheit (22.2° Celsius); this is the optimal operating temperature for your CCU. Ensure temperatures never exceed 85° Fahrenheit (28° Celsius).

Although each CCU chassis is completely shielded from static electric charges, other items such as cables may not be. And since low humidity increases the chances the CCU may fail due to an inadvertent static electricity charge, try to maintain an optimal relative humidity (about 50 percent) in the server room. Too much humidity can be bad as well, so relative humidity should never exceed 95 percent.

Even if you maintain proper humidity, always use good grounding procedures when cabling or servicing the interior of a CCU. In low-humidity areas, always ground yourself, using a grounding strap, when plugging in cables or servicing a CCU.

Dust can clog CCU ventilation filters; the server room should be as dust-free as possible. In addition, clean your CCUs' fan filters on the back of the units periodically.

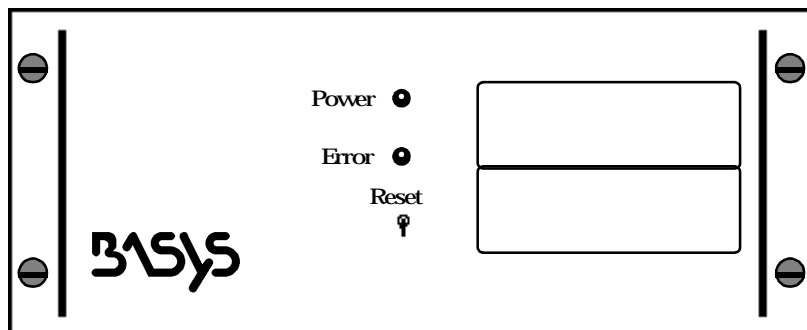
To remove a filter of a CCU II or CCU III:

1. Unscrew four screws in corners of the filter holder.
2. Snap out plastic piece on the back.
3. Remove foam filter.
4. Wash filter with warm water, and completely dry it before reinstallation.
5. Although filters on top and bottom of the chassis are less likely to get dirty, check them occasionally and clean them if necessary. (These filters may be vacuumed.)

## CCU II

The CCU II is no longer supported by Avstar NRCS. This section is included solely for legacy systems, and describes the parts of a CCU II (also known as an AT CCU) with which you need to be familiar to do normal system maintenance work on existing systems. Most importantly, you learn how to reset the CCU, how to connect devices to it, and how to connect it to a server.

## Resetting a CCU II



**Figure D-3** CCU II Front Panel

To reset a CCU II:

1. Pull up on the reset switch and then let go.

This causes the CCU to test its hardware and then boot up to a point where you can restart it from the console.

While the CCU tests itself, it turns on the red error light. This light remains on for several seconds until the CCU completes its self test.

2. When the error light goes out, the CCU has booted successfully and you can restart it from the console.

If the error light does not go out, the CCU has detected a hardware problem. Sometimes these problems are transitory, and resetting the CCU may fix it. However, if the red light does not go out after you reset the CCU a second time, call Avstar Customer Support for assistance.

### Locating Panel Lights

The CCU II's front panel contains a green power light, a red error light, and a reset switch. The green power light is on while the CCU is turned on. If this light goes out, check the power cord to make sure it is firmly connected to both the CCU and power supply.

The red error light indicates whether the CCU is operational. It comes on when the CCU detects a hardware error; it also remains on while the CCU runs through its hardware tests after you have reset it.

## Connecting Devices to a CCU II

Use the I/O ports to connect serial devices, such as printers and wires, to the CCU. These ports are numbered 1 through 8. Use these port numbers in combination with the CCU device number when assigning device numbers to devices you connect to the CCU.

These ports are Data Terminal Equipment (DTE) RS-232 serial ports.

### Locating Ports

The back panel of a CCU II contains the CCU I/O ports, power receptacle, and an Ethernet connector.

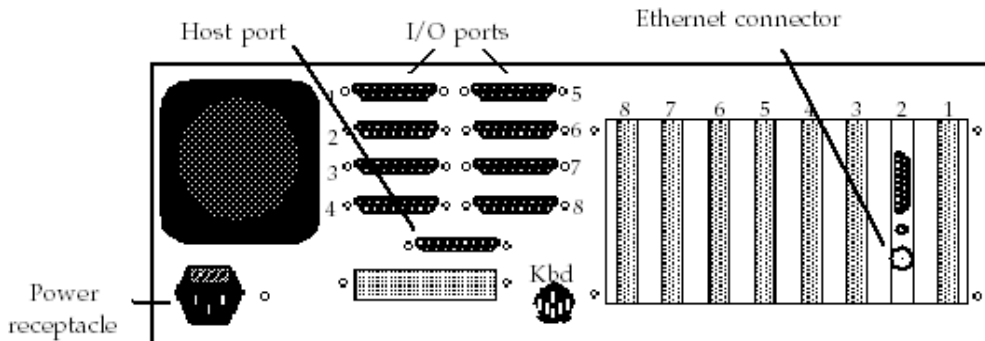
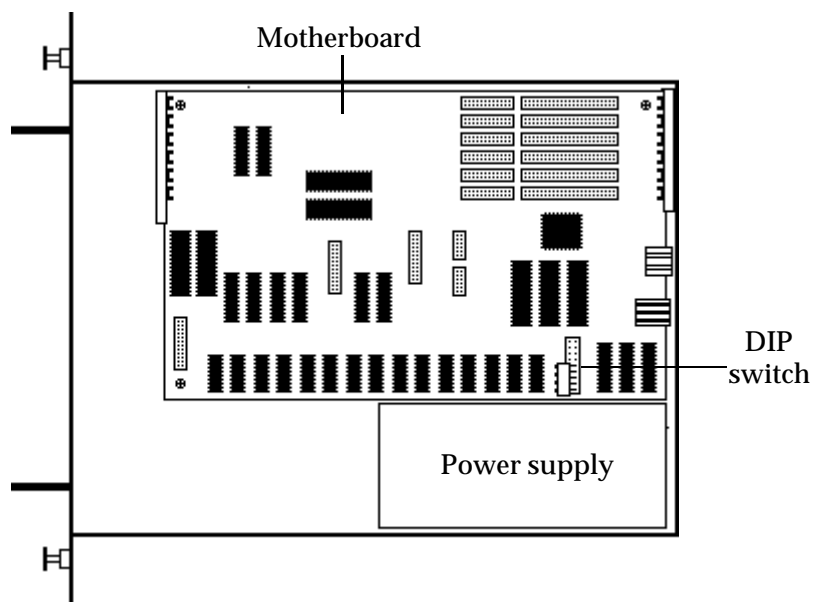


Figure D-4 CCU II Back Panel

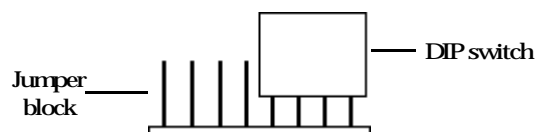
## Connecting a CCU to the System

The CCU II communicates with the system over the network. Network communication is enabled by a DIP switch, as shown in Figure D-5.



**Figure D-5** CCU II with the Top off

The switch is connected to a jumper block (two rows of eight vertical pins each). Seen head on, the switch looks like the ones in Figure D-6.



**Figure D-6** Host Baud Rate DIP Switch

The DIP switch consists of four slider switches that you move up to turn on and down to turn off. When a switch is on, it is in the up position. When it is off, it is in the down position. These switches are very small, so use a pen or pencil point to move them.



*To enable network communication, all switches should be off.*

## CCU III

This section describes parts of a CCU III (or CCU III-A) that you need to be familiar with to do normal system maintenance work. It includes the following tasks:

- Resetting the CCU
- Connecting the CCU to a server
- Connecting devices to the CCU

This section also discusses replacing an Ethernet card and changing the speed at which the CCU communicates with its server.

### Resetting a CCU III

To reset a CCU III, do the following:

1. Pull the reset switch up and let go.

The CCU tests its hardware and, if the hardware is okay, boots up to a point where you can restart it from the console.

As the CCU tests itself, it shows a number of status codes on the LED display. If the CCU passes all its diagnostic tests, it displays a three-digit code indicating that it has successfully completed the self test and is ready to be restarted. Network CCUs display “000” after resetting successfully.

2. If the CCU finds a hardware error, it displays an error code (some number other than “000”) indicating the type of error. Sometimes a hardware error is a transitory one, and resetting the CCU a second time may cause the CCU to reset without error.
3. If the CCU continues to display the error code after you reset it a second time, note the error code and call Avstar Customer Support for assistance.



## Locating Panel Lights

As shown in Figure D-7, a CCU III's front panel contains an LED display and a reset switch. The reset switch clears the CCU memory prior to restarting it. The LED display allows the CCU to display its current status.

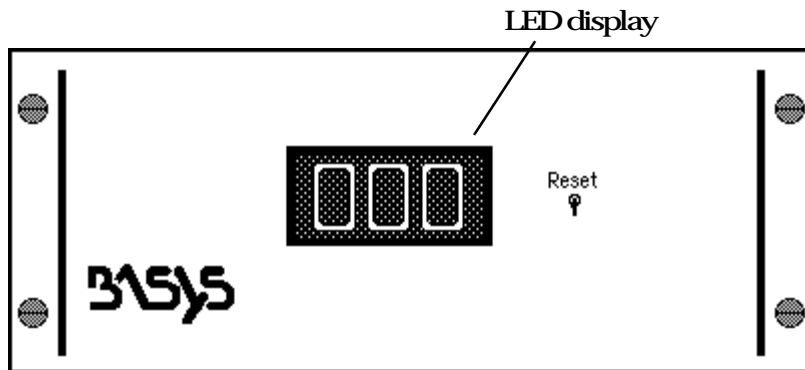


Figure D-7 CCU III Front Panel

## Understanding the LED Display

The CCU uses the LED display to indicate whether it is running normally, has just been reset, or has detected a hardware error. As long as the CCU is running normally, the LED displays the CCU device number divided by 10. For instance, CCU 50 displays "005" and CCU 110 displays "011." The CCU cannot directly display its device number because, while device numbers may be four digits long, the LED can only display three digits. Dividing the device number by 10 is the only way the CCU can display its device number.

If you have a CCU III-A, CCU-IV, or CCU-IVA, its LED has four digits, but the device number is still truncated as described.

If the LED display goes out, the CCU may have lost power; do the following:

1. Make certain the power switch is in the “on” position.
2. Make certain the CCU power cord is firmly attached to the CCU and power supply.

## POST Error Messages

When you reset a CCU III, the codes in Table D-2 appear in succession on the CCU LED display as the CCU tests different parts of its hardware.

**Table D-2 POST Error Messages**

Hex	Dec	Test
1	1	Processor test 1: Processor status verification
2	2	Determine type of post test
3	3	Clear 8042 interface
4	4	Reset 8042
5	5	Get 8042 manufacturing status
6	6	Init chips (DMA, 8259's, CMOS)
7	7	Processor test #2
8	8	Initialize CMOS chip
9	9	EPROM checksum for 32 KB
A	10	Initialize video interface
B	11	Test 8254 channel 0
C	12	Test 8254 channel 1
D	13	Test 8254 channel 2
E	14	Test CMOS date and timer
F	15	Test CMOS shutdown byte

**Table D-2 POST Error Messages (Continued)**

Hex	Dec	Test
10	16	Test DMA channel 0
11	17	Test DMA channel 1
12	18	Test DMA page registers
13	19	Test 8741 keyboard controller
14	20	Test memory refresh toggle circuits
15	21	Test first 64 KB of system memory
16	22	Set up interrupt vector table
17	23	Set up video I/O operations
18	24	Test video memory
19	25	Test 8259 channel-1 mask bits
1A	26	Test 8259 channel-2 mask bits
1B	27	Test CMOS battery level
1C	28	Test CMOS checksum
1D	29	Set up configuration byte from CMOS
1E	30	Size system memory and compare with CMOS
1F	31	Test found system memory
20	32	Test stuck 8259's interrupt bits
21	33	Test stuck NMI (parity/IO check) bits
22	34	Test 8259 interrupt functionality
23	35	Test protected mode and A20 gate
24	36	Sizing extended memory above 1 MB
25	37	Test found system/extended memory
26	38	Test exceptions in protected mode
27	39	Reserved

**Table D-2 POST Error Messages (Continued)**

Hex	Dec	Test
28	40	Reserved

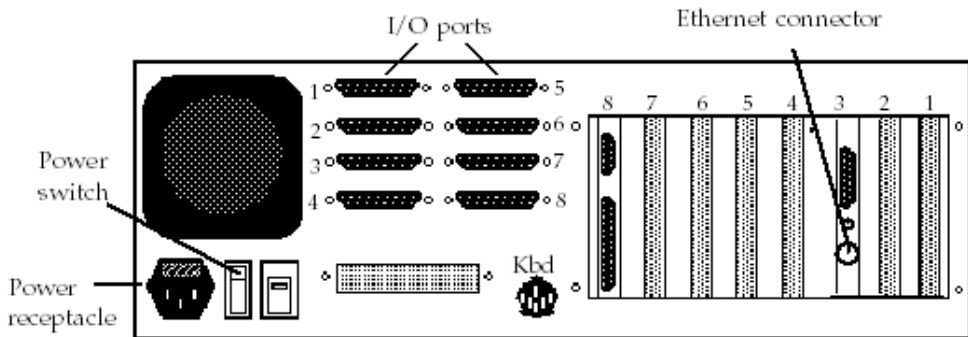
## Connecting Devices to a CCU III

Use I/O ports to connect serial devices such as printers and wires to the CCU. These ports are numbered 1 through 8. Use these port numbers in combination with the CCU device number when assigning device numbers to the devices you connect to the CCU.

These ports are Data Terminal Equipment (DTE) RS-232 serial ports.

## Locating Ports

As shown in Figure D-8, the CCU back panel contains eight I/O ports, a power receptacle, power switch, and an Ethernet connector.



**Figure D-8 CCU III Back Panel**

The power switch controls power to the CCU; it is always in the “on” position. The power receptacle is where the power cord attaches to the CCU.

## Connecting a CCU to the System

A CCU III has an Ethernet connector like the one shown in Figure D-8. Use this connector to attach the CCU to the Ethernet network so it can communicate with its server.

### CCU III Backplane

The CCU III uses a common backplane design, so all CCU electronic components can be on plug-in cards that attach to a single board, or backplane. This modular design facilitates repair of failed components and makes upgrading or expanding the CCU a simple matter of unplugging one card and replacing it with a new one. The electronic components in a CCU III are arranged on separate boards that plug into the backplane. For a network CCU, there are three boards connected to the backplane (see Figure D-9).

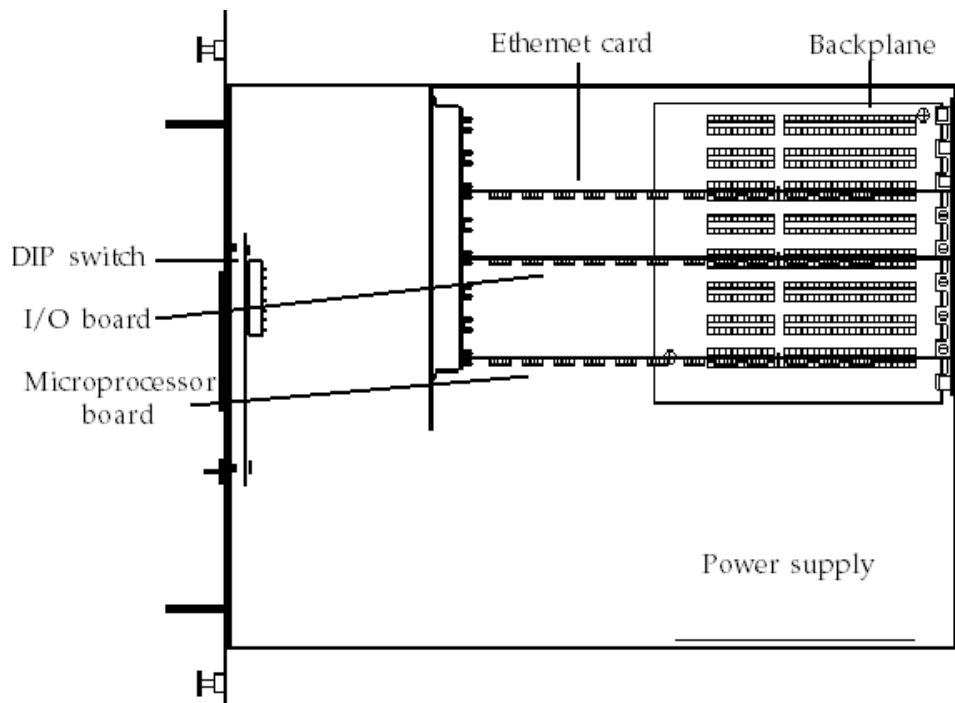


Figure D-9 CCU III with the Top Removed

CCU IIIs have a microprocessor board, which is usually in the eighth slot. This board contains the CCU processor and memory and is the main component of your CCU. Every CCU III also has an I/O board (usually located in the fifth slot) that enables the CCU to communicate with its devices.

CCU IIIs have an Ethernet board, which is usually located in the third slot. This board allows the CCU to use the Ethernet network to communicate with your system's servers.

### Setting the Host Baud Rate DIP Switch

The CCU III communicates with the system over the network. To enable that communication, the DIP switch must be configured as in Figure D-10. Switches one and two are "on," and switches three through eight are "off."

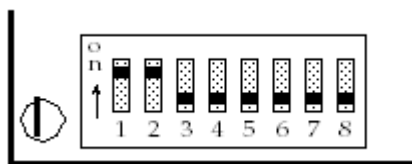


Figure D-10 Host Baud Rate DIP Switch

## CCU IV

A CCU IV is a pair of independent CCU IIIs in one housing. This section describes parts of a CCU IV with which you need to be familiar to do normal system maintenance work. Most importantly, this section includes how to configure and reset the CCU, how to connect the CCU to a server, and how to connect devices to the CCU.

---

## Resetting a CCU IV

To reset a CCU IV, do the following:

1. Press the appropriate Reset button. The CCU tests its hardware and, if the hardware is okay, boots up to a point where you can restart it from the console.

As the CCU tests itself, it shows a number of status codes on the LED display. If the CCU passes all its diagnostic tests, it displays “000,” indicating that it has successfully completed the self test and is ready to be restarted.

If the CCU finds a hardware error, it displays an error code (some number other than “000,” meaning network reset OK) indicating the type of error.

2. Sometimes a hardware error is transitory, and resetting the CCU a second time may cause the CCU to reset without error.

If the CCU continues to display the error code after you reset it a second time, note the error code and call Avstar Customer Support for assistance.

## Locating Front Panel Lights

A CCU IV’s front panel contains two LED displays, two Reset buttons, and two Power buttons. The Reset buttons “clear” each CCU memory prior to restarting it. The LED displays allow each CCU to display its current status.

The Power buttons cycle the power on each CCU separately; use the Power switch on the back to turn the whole unit on or off.

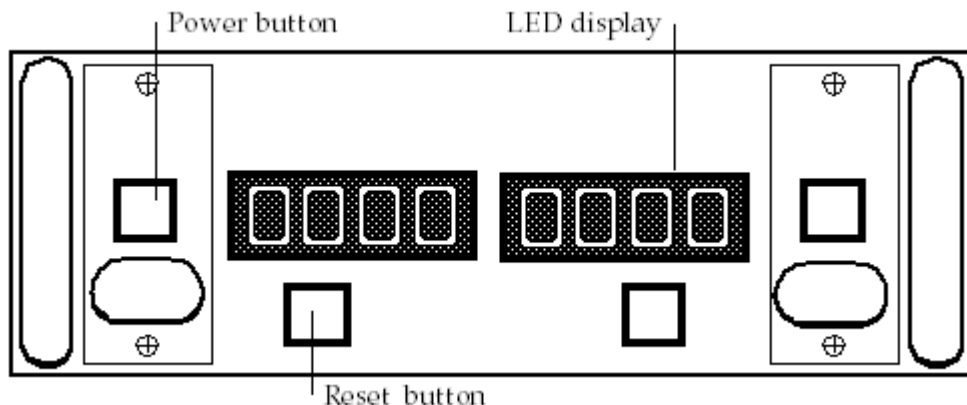


Figure D-11 CCU IV Front Panel

## Understanding the LED Display

The CCU uses the LED display to indicate whether it is running normally, has just been reset, or has detected a hardware error. As long as the CCU is running normally, the LED displays the CCU device number divided by 10. For instance, CCU 50 displays “005” and CCU 110 displays “011.”

The CCU cannot directly display its device number because, while device numbers may be four digits long, the LED can only display three digits. Dividing the device number by 10 is the only way the CCU can display its device number. If you have a CCU IV-A, its LED has four digits, but the last digit of the device number is still truncated.

If the LED goes out, the CCU may have lost power, so:

1. Make certain the power switch is in the “on” position.
2. Make certain the CCU power cord is firmly attached to the CCU and power supply.



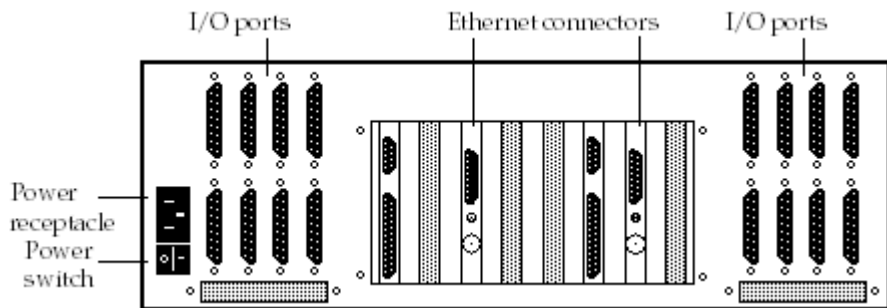
## Connecting Devices to a CCU IV

Each set of I/O ports are numbered 1 through 8. Use these port numbers in combination with the CCU device number when assigning device numbers to devices you connect to the CCU.

These ports are Data Terminal Equipment (DTE) RS-232 serial ports.

### Locating Ports

As shown in Figure D-12, a fully-configured CCU IV's back panel contains sixteen I/O ports, two host ports, two Ethernet connectors, a power receptacle, and a power switch.



**Figure D-12 CCU IV Back Panel**

The power switch controls power to the entire CCU IV; it is always in the “on” position. The power receptacle is where the power cord attaches to the CCU.

## Connecting a CCU IV to the System

A CCU IV has two Ethernet connectors, which are used to attach each CCU to the Ethernet network so it can communicate with its server.

### CCU IV Interior

The CCU IV uses a common backplane design, so all CCU electronic components can be on plug-in cards. This modular design facilitates the repair of failed components and makes upgrading or expanding the CCU a simple matter of unplugging one card and replacing it with a new one. Figure D-13 shows how the electronic components in a CCU IV are arranged on separate boards that plug into the backplane. There are three boards connected to the backplane for each CCU you have set up, for a possible total of six.

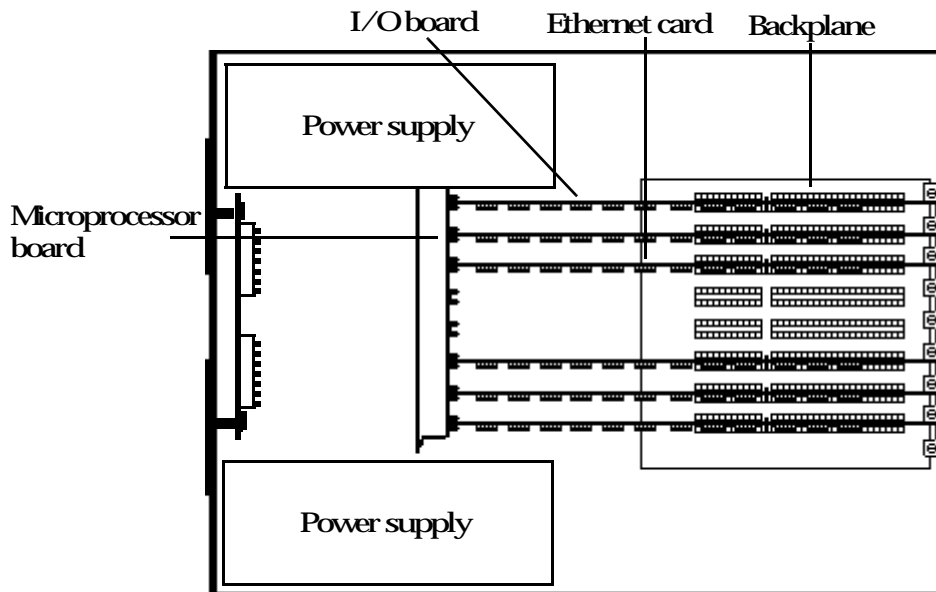


Figure D-13 CCU IV with the Top Removed

CCU IVs have two microprocessor boards, which contain the CCUs' processors and memory. Every CCU IV also has two I/O boards to enable each CCU to communicate with its devices.

CCU IVs have two Ethernet boards, which allow each CCU to use the Ethernet network to communicate with your system's servers.

## PCU

There are two kinds of PCUs: older models in rack-mount cases and newer models in standard PC cases. This section describes the rack-mount models.

A PCU is essentially a DOS PC designed to function just like a network CCU (either a CCU III or IV). This section describes parts of a PCU with which you need to be familiar to do normal system maintenance work. It also describes how to reset the PCU, how to connect the PCU to a server, and how to connect devices to the PCU.



*Your system may have existing CCUs that have been upgraded to PCUs. Such devices must be configured as PCUs, although their physical operation remains the same.*

## Resetting a PCU

You may occasionally need to reset a PCU to clear its memory so that you can load a fresh set of programs. To do this, press the Reset button and let go. The PCU tests its hardware and boots up to a point where you can restart it from the Avstar console.

As the PCU tests itself, it shows a number of status codes on the LED display. When DOS boots on the PCU, it displays "8000." If the PCU passes all its diagnostic tests and receives an IP address from a `bootp` server, it displays "0000," indicating that it has successfully completed the self test and is ready to be restarted. If the PCU continues to display "8000," there is a network or `bootp` problem.

A PCU displays “8888” after receiving a reset request from the host server. Once it has been restarted and is operational, it displays its device number (the device number is not divided by 10).

### **Locating Front Panel Lights**

A PCU can look just like a CCU III or CCU IV, or it can be an ordinary PC with special additional hardware to give it more serial ports.

If it has a CCU-style housing, the PCU’s front panel contains one or two LED displays and reset switches. The reset switches allow you to clear the PCU’s memory prior to restarting it. LEDs display the PCU’s device number or current status.

### **Understanding the LED Display**

A PCU built to resemble a CCU has one or two LED displays with four digits each. It uses the LED display to indicate whether it is running normally, has just been reset, or has detected an error. As long as the PCU is running normally, the LED displays the PCU’s device number.

If the LED goes out, the PCU may have lost power, so make certain the power switch is in the “on” position and the PCU’s power cord is firmly attached to the PCU and power supply. PCUs produced in regular PC housings do not have LED displays.

### **PCU LED Codes**

These are the standard codes that can appear on the PCU’s LED display. The 8000 codes are typically shown during a normal boot of the PCU, while the 9000 codes represent errors. Contact Avstar Customer Support for assistance if one of your PCUs displays an error code after being restarted.

**Table D-3 PCU LED Codes**

<b>Code</b>	<b>Meaning</b>
0000	Network initialization complete
8000	Boot started
8002	Connected to host server for download of PCU OS
8003	PCU OS download in progress
8004	PCU OS downloaded
8005	Start received
8006	Invoking PCU OS
8100	PCU OS started
8101	PCU OS network initialized
8102	PCU OS fully operational
8888	Reset received
9001	Error, execv failed
9002	Error, file could not be opened
9003	Error, no data downloaded
9004	Error, cannot establish socket connection
9005	Error, read on socket failed
9006	Error, receive on socket failed
9007	Error, send on socket failed
9010	Memory management—bad free
9011	Memory management—unknown free
9012	Memory management—no space

## Connecting Devices to a PCU

You use the I/O ports to connect serial devices, such as printers and wires, to the PCU. These ports are numbered 1 through 8. Use these port numbers in combination with the PCU's device number when assigning device numbers to devices you connect to the PCU. These ports are Data Terminal Equipment (DTE) RS-232 serial ports. To connect a device to one of these ports, use a cable built to carry signals the device and PCU need, in a way they expect.

### Locating Ports

The back panel of a PCU built in a CCU III housing contains eight I/O ports, a host port, a power receptacle, an Ethernet connector, and a power switch.

The power switch controls power to the PCU and should always be in the "on" position. The power receptacle is where the power cord attaches to the PCU. The host port is used to attach a diagnostic workstation to the PCU, if necessary.

A PCU in a standard PC case looks similar to this, except that eight serial I/O ports are on a serial card in one of the PC's expansion slots.

## Connecting a PCU to the System

A PCU has an Ethernet connector. Use this connector to attach the PCU to the Ethernet network so it can communicate with its server.

# APPENDIX E

## *Character Mapping Code Tables for Wires*

This appendix contains character mapping code tables used by wire programs. The value set in the “bits” wire profile option determines which of these tables the wire program uses to “translate” stories it receives from a wire service. Unless otherwise indicated, the tables contain, in the first column, decimal numbers sent by the wire service. Also, the other columns contain characters represented by those numbers, descriptions of the characters, and mapped values.

Also included in this appendix are examples of character mapping for Arabic wires.

The major sections in this appendix are:

- ASCII (7-bit) Character Set
- IBM Character Set
- dbrestore Conversion Map (Arabic)
- Sample Arabic Wire Profile

## ASCII (7-bit) Character Set

This table is an identity mapping, where the wire value equals the mapped value, with one exception—the wire value of 124 is mapped to 0. The other 127 values are preserved.

**Table E-1**     **ASCII (7-bit)**

<b>Value</b>	<b>Mapped Alias</b>	<b>Description</b>	<b>Mapped Value</b>
000	<nul>	null	000
001	<soh>	start of header	001
002	<stx>	start of text	002
003	<etx>	end of text	003
004	<eot>	end of transmission	004
005	<enq>	enquiry	005
006	<ack>	acknowledgment	006
007	<bel>	bell	007
008	<bs>	back space	008
009	<ht>	horizontal tab	009
010	<lf>	line feed	010
011	<vt>	vertical tab	011
012	<ff>	form feed	012
013	<cr>	carriage return	013

---



**Table E-1 ASCII (7-bit) (Continued)**

<b>Value</b>	<b>Mapped Alias</b>	<b>Description</b>	<b>Mapped Value</b>
014	<so>	shift out	014
015	<si>	shift in	015
016	<dle>	data link escape	016
017	<dcl>	device control 1	017
018	<dc2>	device control 2	018
019	<dc3>	device control 3	019
020	<dc4>	device control 4	020
021	<nak>	negative acknowledge	021
022	<syn>	synchronous idle	022
023	<etb>	end of transmission	023
024	<can>	cancel	024
025	<em>	end of medium	025
026	<sub>	substitute	026
027	<esc>	escape	027
028	<fs>	file separator	028
029	<gs>	group separator	029
030	<rs>	record separator	030
031	<us>	unit separator	031
032	<sp>	space	032
033	!		033
034	"		034

**Table E-1    ASCII (7-bit) (Continued)**

<b>Value</b>	<b>Mapped Alias</b>	<b>Description</b>	<b>Mapped Value</b>
035	#		035
036	\$		036
037	%		037
038	&		038
039	'	quote	039
040	(		040
041	)		041
042	*		042
043	+		043
044	,	comma	044
045	-		045
046	.	period	046
047	/		047
048	0		048
049	1		049
050	2		050
051	3		051
052	4		052
053	5		053
054	6		054
055	7		055

**Table E-1 ASCII (7-bit) (Continued)**

<b>Value</b>	<b>Mapped Alias</b>	<b>Description</b>	<b>Mapped Value</b>
056	8		056
057	9		057
058	:	colon	058
059	<sc>		059
060	<		060
061	=		061
062	>		062
063	?		063
064	@		064
065	A		065
066	B		066
067	C		067
068	D		068
069	E		069
070	F		070
071	G		071
072	H		072
073	I		073
074	J		074
075	K		075
076	L		076

**Table E-1 ASCII (7-bit) (Continued)**

<b>Value</b>	<b>Mapped Alias</b>	<b>Description</b>	<b>Mapped Value</b>
077	M		077
078	N		078
079	O		079
080	P		080
081	Q		081
082	R		082
083	S		083
084	T		084
085	U		085
086	V		086
087	W		087
088	X		088
089	Y		089
090	Z		090
091	[		091
092	\ or<bks>		092
093	]		093
094	^		094
095	_	underscore	095
096	'		096
097	a		097

**Table E-1 ASCII (7-bit) (Continued)**

<b>Value</b>	<b>Mapped Alias</b>	<b>Description</b>	<b>Mapped Value</b>
098	b		098
099	c		099
100	d		100
101	e		101
102	f		102
103	g		103
104	h		104
105	i		105
106	j		106
107	k		107
108	l		108
109	m		109
110	n		110
111	o		111
112	p		112
113	q		113
114	r		114
115	s		115
116	t		116
117	u		117
118	v		118

**Table E-1 ASCII (7-bit) (Continued)**

Value	Mapped Alias	Description	Mapped Value
119	w		119
120	x		120
121	y		121
122	z		122
123	{		123
124	<nul>		0
125	}		125
126	~		126
127	<del>	Delete	127

## IBM Character Set

The IBM character set is the same as the ASCII 7-bit character set for the first 127 characters. For the remaining characters, see the table below.

**Table E-2 IBM**

Value	Wire Value HEX	Mapped Alias	Mapped Value	Mapped Value HEX
128	80	Ç	190	C7
129	81	ü	252	FC
130	82	é	233	E9
131	83	â	226	E2
132	84	ä	228	E4

**Table E-2 IBM**

<b>Value</b>	<b>Wire Value HEX</b>	<b>Mapped Alias</b>	<b>Mapped Value</b>	<b>Mapped Value HEX</b>
133	85	à	224	E0
134	86	á	229	E5
135	87	ç	231	E7
136	88	ê	234	EA
137	89	ë	235	EB
138	8A	è	232	E8
139	8B	ï	239	EF
140	8C	î	238	EE
141	8D	ì	236	EC
142	8E	Ä	196	C4
143	8F	Å	197	C5
144	90	É	201	C9
145	91	æ	230	E6
146	92	Æ	198	C6
147	93	ô	244	F4
148	94	ö	246	F6
149	95	ò	242	F2
150	96	û	251	FB
151	97	ù	249	F9
152	98	ÿ	253	FD
153	99	Ö	214	D6

**Table E-2 IBM**

<b>Value</b>	<b>Wire Value HEX</b>	<b>Mapped Alias</b>	<b>Mapped Value</b>	<b>Mapped Value HEX</b>
154	9A	Ü	220	DC
155	9B	ç	162	A2
156	9C	£	163	A3
157	9D	¥	165	A5
158	9E	<nul>	0	0
159	9F	<nul>	0	0
160	A0	á	225	E1
161	A1	í	237	ED
162	A2	ó	243	F3
163	A3	ú	250	FA
164	A4	ñ	241	F1
165	A5	Ñ	209	D1
166	A6	ª	170	AA
167	A7	º	186	BA
168	A8	<nul>	0	0
169	A9	<nul>	0	0
170	AA	<nul>	0	0
171	AB	<nul>	0	0
172	AC	<nul>	0	0
173	AD	<nul>	0	0
174	AE	<nul>	0	0



**Table E-2 IBM**

<b>Value</b>	<b>Wire Value HEX</b>	<b>Mapped Alias</b>	<b>Mapped Value</b>	<b>Mapped Value HEX</b>
175	AF	<nul>	0	0
176	B0	<nul>	0	0
177	B1	<nul>	0	0
178	B2	<nul>	0	0
179	B3	<nul>	0	0
180	B4	<nul>	0	0
181	B5	<nul>	0	0
182	B6	<nul>	0	0
183	B7	<nul>	0	0
184	B8	<nul>	0	0
185	B9	<nul>	0	0
186	BA	<nul>	0	0
187	BB	<nul>	0	0
188	BC	<nul>	0	0
189	BD	<nul>	0	0
190	BE	<nul>	0	0
191	BF	<nul>	0	0
192	C0	<nul>	0	0
193	C1	<nul>	0	0
194	C2	<nul>	0	0
195	C3	<nul>	0	0

**Table E-2 IBM**

<b>Value</b>	<b>Wire Value HEX</b>	<b>Mapped Alias</b>	<b>Mapped Value</b>	<b>Mapped Value HEX</b>
196	C4	<nul>	0	0
197	C5	<nul>	0	0
198	C6	<nul>	0	0
199	C7	<nul>	0	0
200	C8	<nul>	0	0
201	C9	<nul>	0	0
202	CA	<nul>	0	0
203	CB	<nul>	0	0
204	CC	<nul>	0	0
205	CD	<nul>	0	0
206	CE	<nul>	0	0
207	CF	<nul>	0	0
208	D0	<nul>	0	0
209	D1	<nul>	0	0
210	D2	<nul>	0	0
211	D3	<nul>	0	0
212	D4	<nul>	0	0
213	D5	<nul>	0	0
214	D6	<nul>	0	0
215	D7	<nul>	0	0
216	D8	<nul>	0	0

**Table E-2 IBM**

<b>Value</b>	<b>Wire Value HEX</b>	<b>Mapped Alias</b>	<b>Mapped Value</b>	<b>Mapped Value HEX</b>
217	D9	<nul>	0	0
218	DA	<nul>	0	0
219	DB	<nul>	0	0
220	DC	<nul>	0	0
221	DD	<nul>	0	0
222	DE	<nul>	0	0
223	DF	<nul>	0	0
224	E0	<nul>	0	0
225	E1	β	223	DF
226	E2	<nul>	0	0
227	E3	<nul>	0	0
228	E4	<nul>	0	0
229	E5	<nul>	0	0
230	E6	μ	185	B5
231	E7	<nul>	0	0
232	E8	<nul>	0	0
233	E9	<nul>	0	0
234	EA	<nul>	0	0
235	EB	<nul>	0	0
236	EC	<nul>	0	0
237	ED	∅	248	F8

**Table E-2 IBM**

<b>Value</b>	<b>Wire Value HEX</b>	<b>Mapped Alias</b>	<b>Mapped Value</b>	<b>Mapped Value HEX</b>
238	EE	<nul>	0	0
239	EF	<nul>	0	0
240	F0	<nul>	0	0
241	F1	±	177	B1
242	F2	<nul>	0	0
243	F3	<nul>	0	0
244	F4	<nul>	0	0
245	F5	<nul>	0	0
246	F6	<nul>	0	0
247	F7	<nul>	0	0
248	F8	°	176	B0
249	F9	<nul>	0	0
250	FA	<nul>	0	0
251	FB	<nul>	0	0
252	FC	<nul>	0	0
253	FD	<sup>2</sup>	178	B2
254	FE	<nul>	0	0
255	FF	<nul>	0	0

## dbrestore Conversion Map

Here is an example of a `dbrestore` conversion map—in this case, converting characters from ASMO 449+ to Arabic Windows 1256:

```
#this file maps from ASMO 449+ to MS 1256 code page
130 -> 233           ;e acute
131 -> 226           ;a circumflex
133 -> 224           ;a grave
135 -> 231           ;c cidilla
136 -> 234           ;e circumflex
137 -> 235           ;e umlaut
138 -> 232           ;e grave
139 -> 239           ;i umlaut
140 -> 238           ;i circumflex
147 -> 244           ;o circumflex
150 -> 251           ;u circumflex
151 -> 249           ;u grave
172 -> 161           ;arabic comma
174 -.>171           ;left double angle quote
175 -> 187           ;right double angle quote
187 -.>186           ;arabic semi colon
215 - 218 -> 216 - 219 ;tah,zah,ain,ghain
224 - 227 -> 220 - 223 ;tatweel,feh,qaf,kaf
228 -> 225           ;lam
229 - 232 -> 227 - 230 ;meem, noon, heh, waw
233 - 234 -> 236 - 237 ;alef maksura, yeh
235 - 238 -> 240 - 243 ;fathatan,dammatan,kasratan,fatha
239 - 240 -> 245 - 246 ;damma, kasra
241 -> 248           ;shadda
242 ->250           ;sukun
246 -> 161           ;arabic comma
247 => 225 194       ;lam alef with madda above
248 => 225 195       ;lam alef with hamza above
249 => 225 197       ;lam alef with hamza below
250 => 225 199       ;lam alef
# that s all
```

## Sample Arabic Wire Profile

Here is an example of a wire profile for an Arabic wire service, using the `map` and `reverse` wire profile options.

```
wire 11 300-5 dummy MIN - - ;MINA service

# cat /site/wires/11
; MENA WIRE
;AVSTAR 1.2 MS Codepage 1256

form wire-arabic
bits 5
flags pblines
start
<212><237><212><237><128><LF><128><LF><128><LF><128><LF>
<128><LF><128><LF>
reverse 0123456789 /,.
end <228><228><228>
slug 30
accent <bs>
form wire
flags addheol

map <00> _<bs>
map <01> <216>
map <02> <CR><LF>
map <03> <213>
; 4 is space
map <05> <198>
map <06> <200>
map <07> <203>
map <08> <223>
map <09> <222>
map <10> <207>
map <11> <229>
map <12> <228>
map <13> <221>
```

```
map          <14>    <237>
map          <15>    <205>
map          <16>    <230>
map          <17>    <212>
map          <18>    <204>
map          <19>    <225><199>
map          <20>    <218>
map          <21>    <209>
map          <22>    <195>
map          <23>    <225>
map          <24>    <202>
map          <25>    <199>
map          <26>    <218>
;27 is figure shift
map          <28>    <227>
map          <29>    <211>
map          <30>    <199><225>
;31 is letter shift
map          <32>    _<bs>
map          <33>    <55>
map          <34>    <CR><LF>
;34 is new line
map          <35>    <213>
;36 is space
map          <37>    <236>
map          <38>    <50>
map          <39>    <51>
map          <40>    <223>
map          <41>    <222>
map          <42>    <54>
map          <43>    <229>
map          <44>    <228>
map          <45>    <201>
map          <46>    <236>
map          <47>    <205>
map          <48>    <53>
map          <49>    <212>
map          <50>    <204>
```

```
map          <51>    <56>
map          <52>    <47>
map          <53>    <52>
map          <54>    <48>
map          <55>    <57>
map          <56>    <49>
map          <57>    <193>
map          <58>    <218>
;59 is figure shift
map          <60>    <227>
map          <61>    <211>
map          <62>    <225>
;63 is letter shift
```



# APPENDIX F

## *Environment Variables*

Some features in Avstar NRCS require the system administrator to set up environment variables in the Registry of the workstations. The person responsible for setting up these variables should have a good understanding of Windows<sup>®</sup>-based operating systems, and the Registry Editor program. This appendix includes the following sections:

- Registry Editor
- Environment Variables
  - CCColor
  - DestinationOrder
  - MailLookup
  - MsgMailAlert
  - PIColor
  - ShowTimingBar
  - SyncToServer
  - VT Compatibility
  - Delete\_Notify

## Registry Editor

The Registry Editor is used to create and define environment variables (Registry values) at each workstation.

To access the Registry Editor, do the following:

1. Click the Start button on the Windows Taskbar.
2. Select the Run option.
3. Type `regedit` in the dialog box that appears.
4. The Registry Editor window will appear. See Figure F-1 “Registry Editor Window” on page F-4.

All Avstar NRCS environment variables are set up and stored in the same location on each workstation. After you open the Registry Editor window, navigate to the following folder (also called a key):

```
HKEY_LOCAL_MACHINE\  
    SYSTEM\  
        CurrentControlSet\  
            Control\  
                Session Manager\  
                    Environment
```



*On workstations running the Windows NT<sup>®</sup> operating system, there are two keys with similar names: `SessionManager` and `Session Manager`. The one called `Session Manager` (with a space between the two words) should be used.*

On PCs running the Windows<sup>®</sup> 95 operating system, if the `Session Manager` key (with the space) does not exist, it must be created, along with the `Environment` key, before completing instructions for setting up the environment variables.

To create a new key, do the following:

1. Right-click on the key—that is, the folder in the Directory panel of the Registry Editor window—in which you want to create the new key.
2. Choose New key.
3. Type the name of the new key, such as **Session Manager**, or **Environment**.

## Environment Variables

Environment variables are sometimes required to set up certain Avstar features at various Avstar Workstations. Environment variables are located and defined in the Registry of Avstar Workstations—that is, Windows-based PCs running the Avstar Client software.

The following sections explain how you can set up environment variables by manually editing the Registry using the Registry Editor. See “Registry Editor” on page F-2 for more information.



*Self-importing files, called reg files, can be executed to automatically import environment variable information into the Registry. These files with their exported registry keys, can be used on PCs running Windows® 95, 98, NT and 2000 operating systems. For more information on how to obtain and use these reg files, contact iNews Customer Support.*

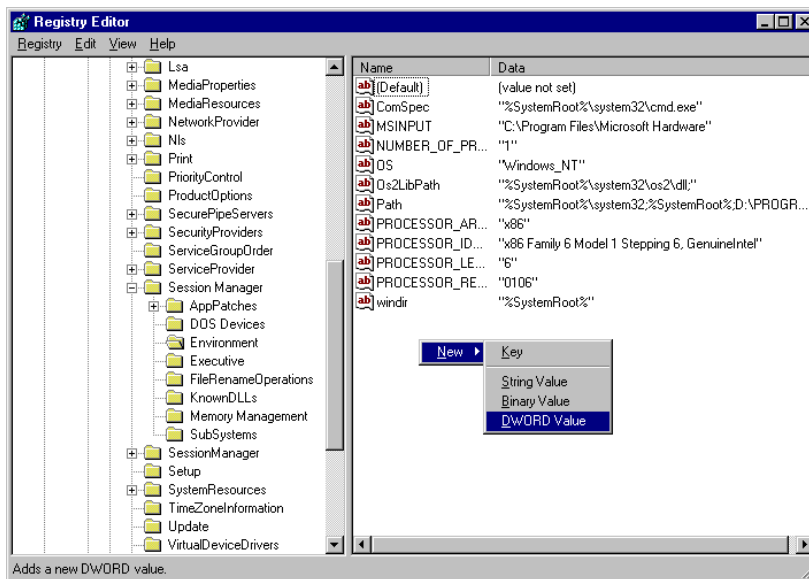
## CCColor

An individual workstation can now have its closed captioning text color changed via an environment variable called `CCColor`. If no environment variable exists, then the default color of green is used.

To change the closed captioning text color, do the following:

1. Open Registry Editor.

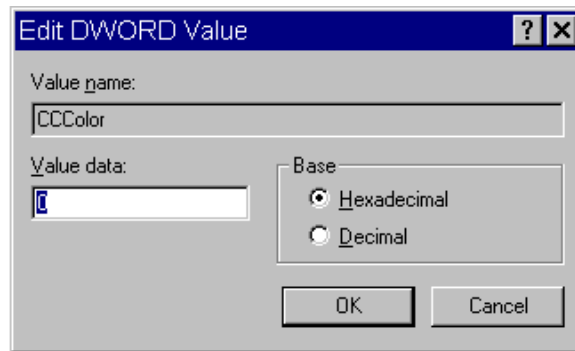
2. Navigate to the Environment key and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu will appear, as shown in Figure F-1.



**Figure F-1 Registry Editor Window**

4. Select the DWORD Value option to create and define a new Registry value of type DWORD in the Registry Editor.
5. Type the name of the new value: CCCo1or.
6. Set the Value data option by doing the following:
  - a. Right-click on the CCCo1or value.
  - b. Choose Modify from the pop-up menu.

The Edit DWORD Value dialog box will appear.



- c. Set the Value data using the hexadecimal format:  
0x00RRGGBB, where RR, GG, BB are two bytes for each color.



*The leftmost two bytes (00) are not used. Also, If the `CCColor` has its value set to zero (0), the closed captioning text will be black because zero corresponds to the color Black.*

- d. Click OK to save the setting and close the dialog box.

7. Close the Registry Editor window.

See “RGB Hexadecimal Color Chart” on page F-11 for more information on possible colors used in this environment variable.

## DestinationOrder

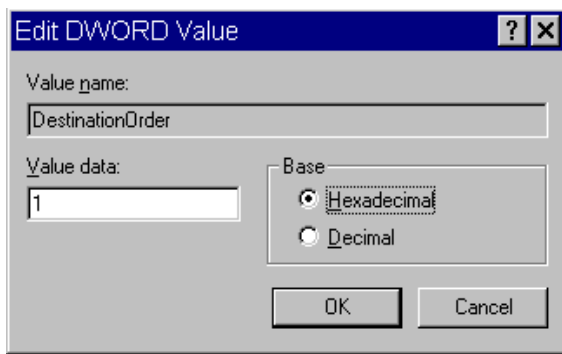
Enabling the destination order feature ensures the user’s Home location is always the top item in the Destination queue list. For instance, when you duplicate a story to another queue, the user’s Home location will always be the top item in the list. It also ensures the user’s Destination location is the second item in the list.

To enable the destination order feature, do the following:

1. Open Registry Editor.

2. Navigate to the `Environment` key, and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu will appear. See Figure F-1 “Registry Editor Window” on page F-4.
4. Select the `DWORD Value` option to create and define a new registry value of type `DWORD` in the Registry Editor.
5. Type the name of the new value: `DestinationOrder`.
6. Set the Value data option by doing the following:
  - a. Right-click on the `DestinationOrder` value.
  - b. Choose `Modify` from the pop-up menu.

The Edit DWORD Value dialog box will appear.



- c. Set the Value data. Type `0` (zero) to disable the destination order feature, or `1` to enable it.



*Any number other than 1 turns `DestinationOrder` off and back to its default behavior, which is to always display the last visited queue/folder as the top item in the destination list.*

- d. Click `OK` to save the setting and close the dialog box.
7. Close the Registry Editor window.

## MailLookup

Avstar NRCS provides users with an e-mail addressee name lookup feature. When used, all groups, aliases, and users that partially match characters in the To: or CC: fields are displayed in a Check Name dialog box for user selection. This is the default behavior. However, system administrators can set an environment variable that defines which matches are displayed for selection in the Check Name dialog box. Consequently, this allows system administrators to hide any groups that exist in the system for reasons other than e-mail purposes.

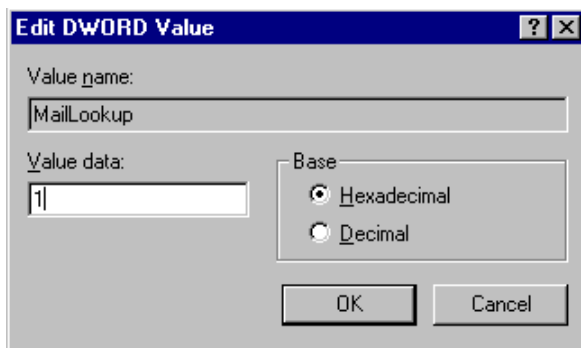


*This environment variable must be created and defined in the Registry Editor at each workstation. Otherwise, default behavior will be used at workstations where the environment variable is not defined.*

To set the environment variable, do the following:

1. Open Registry Editor.
2. Navigate to the `Environment` key, and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu will appear. See Figure F-1 “Registry Editor Window” on page F-4.
4. Select the `DWORD Value` option to create and define a new Registry value of type `DWORD` in the Registry Editor.
5. Type the name of the new value: **MailLookup**.
6. Press Enter.
7. Set the Value data option by doing the following:
  - a. Right-click on the `MailLookup` value.
  - b. Choose `Modify` from the pop-up menu.

The Edit DWORD Value dialog box will appear.



- c. Set the Value data, by typing one of the following four options:

- 0 - (zero) show no matches
- 1 - show only user matches
- 2 - show only group/alias matches
- 3 - show groups/aliases and user matches

The default behavior—without the Registry value `Mail-Lookup` defined at a workstation—is 3.

- d. Click OK to save the setting and close the dialog box.

8. Close the Registry Editor window.

## MsgMailAlert

Enabling the Message Mail Alert feature allows you to change the alert behavior so that Avstar Workstation will flash message and/or mail alerts on the status bar for only 15 seconds, rather than persistently. By adding the `MsgMailAlert` variable, you can specify additional settings.

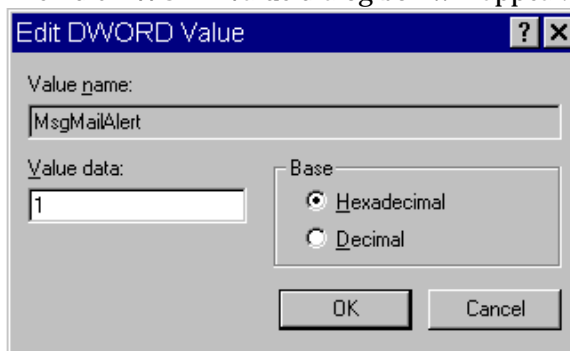
To set the environment variable, do the following:

1. Open Registry Editor.
2. Navigate to the `Environment` key, and open it.



3. Right-click on the right side of the Registry Editor window. A pop-up menu will appear. See Figure F-1 “Registry Editor Window” on page F-4.
4. Select the DWORD Value option to create and define a new Registry value of type `DWORD` in the Registry Editor.
5. Type the name of the new value: `MsgMailAlert`.
6. Press Enter.
7. Set the Value data option by doing the following:
  - a. Right-click on the `MsgMailAlert` value.
  - b. Choose Modify from the pop-up menu.

The Edit DWORD Value dialog box will appear.



- c. Set the Value data, by typing one of the following four options:
  - 0 - disable - no alerts whatsoever on status bar
  - 1 - neither persistent - alerts flash for 15 seconds
  - 2 - only message alerts persistent
  - 3 - only mail alerts persistent
  - 4 - both alerts persistent - alerts will not go away until user has read all correspondence.

The default behavior—without the Registry value `MsgMailAlert` defined at a workstation—is 1.

- d. Click OK to save the setting and close the dialog box.

8. Close the Registry Editor window.

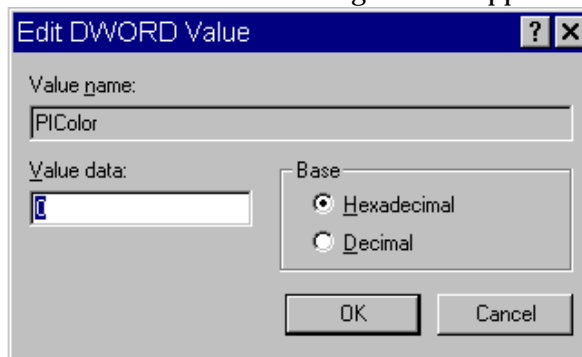
## PIColor

An individual Avstar Workstation can now have its presenter instructions text color changed via an environment variable called `PIColor`. If no environment variable exists, then the default color of red is used.

To change presenter instructions text color, do the following:

1. Open Registry Editor.
2. Navigate to the `Environment` key, and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu will appear. See Figure F-1 “Registry Editor Window” on page F-4.
4. Select the `DWORD Value` option to create and define a new Registry value of type `DWORD` in the Registry Editor.
5. Type the name of the new value: `PIColor`.
6. Set the Value data option by doing the following:
  - a. Right-click on the `PIColor` value.
  - b. Choose `Modify` from the pop-up menu.

The Edit DWORD Value dialog box will appear.



- c. Set the Value data using the hexadecimal format:

0x00RRGGBB, where RR, GG, BB are two bytes for each color.



*The leftmost two bytes (00) are not used. Also, If the PIColor has its value set to zero (0), the closed captioning text will be black because zero corresponds to the color Black.*

- d. Click OK to save the setting and close the dialog box.

7. Close the Registry Editor window.

## RGB Hexadecimal Color Chart

Avstar's PIColor and CCColor environment variables require RGB Hexadecimal Color codes. Here are some basic colors, along with their corresponding hexadecimal code values:

Complete RGB Hexadecimal Color Charts, with various color shades, can be found on the Internet.

<u>Color</u>	<u>Hex</u>	
Black	000000	
Blue	0000FF	
Brown	330000	
Green	008800	(Default color for CCColor)
Orange	FF6600	
Pink	CC0099	
Purple	660099	
Red	FF0000	(Default color for PIColor)
White	FFFFFF	
Yellow	FFFF00	

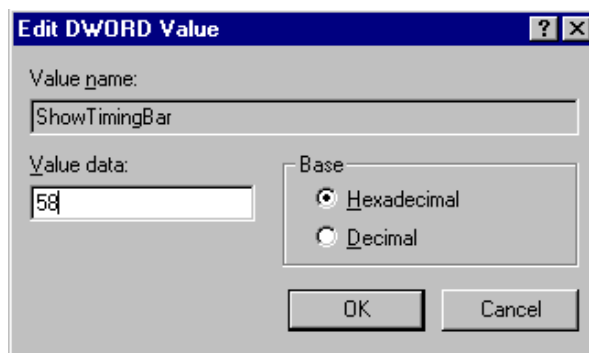
## ShowTimingBar

A system administrator can define which key on the keyboard is used to advance the timing bar during show timing. The default key is the space bar.

To change the setting to a different key, do the following:

1. Open Registry Editor.
2. Navigate to the `Environment` key, and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu will appear. See Figure F-1 “Registry Editor Window” on page F-4.
4. Select the `DWORD Value` option to create and define a new Registry value of type `DWORD` in the Registry Editor.
5. Type the name of the new value: `ShowTimingBar`.
6. Set the Value data option by doing the following:
  - a. Right-click on the `ShowTimingBar` value.
  - b. Choose `Modify` from the pop-up menu.

The Edit DWORD Value dialog box will appear.



The `ShowTimingBar` Value data is determined by the Scan Code of the selected key on the keyboard. For instance, if the system administrator wants to use the F12 key to advance the

timing bar, the Value data for the ShowTimingBar registry would be either the Hexadecimal code of 58 or Decimal code of 88. See the Scan Code table below for more information.

c. Click OK to save the setting and close the dialog box.

7. Close the Registry Editor window.

## Scan Codes

**Table F-1 Scan Codes**

Key	Decimal	Hexadecimal
' ~ (accent/tilde)	41	29
1 ! (exclamation point)	2	02
2 @ (at symbol)	3	03
3 # (pound sign)	4	04
4 \$ (dollar sign)	5	05
5 % (percent)	6	06
6 ^ (carrot)	7	07
7 & (ampersand)	8	08
8 * (asterisk)	9	09
9 ( (open parenthesis)	10	0A
0 ) (close parenthesis)	11	0B
- _ (dash/underscore)	12	0C
= + (equal/plus)	13	0D
Backspace	14	0E
Tab	15	0F
Q	16	10
W	17	11
E	18	12
R	19	13
T	20	14

**Table F-1      Scan Codes**

<b>Key</b>	<b>Decimal</b>	<b>Hexadecimal</b>
Y	21	15
U	22	16
I	23	17
O	24	18
P	25	19
[ { (open bracket/brace)	26	1A
] } (close bracket/brace)	27	1B
Caps Lock	58	3A
A	30	1E
S	31	1F
D	32	20
F	33	21
G	34	22
H	35	23
J	36	24
K	37	25
L	38	26
;; (semicolon/colon)	39	27
' " (accent/quote)	40	28
\   (backslash/pipe)	43	2B
Left Shift	42	2A
Z	44	2C
X	45	2D
C	46	2E
V	47	2F
B	48	30
N	49	31
M	50	32

**Table F-1 Scan Codes**

<b>Key</b>	<b>Decimal</b>	<b>Hexadecimal</b>
, < (comma/less-than)	51	33
. > (period/greater-than)	52	34
/ ? (slash/question mark)	53	35
Right Shift	54	36
CTRL (Control keys)	29	1D
ALT (Alt keys)	56	38
Spacebar	57	39
ESC (Escape key)	1	01
F1	59	3B
F2	60	3C
F3	61	3D
F4	62	3E
F5	63	3F
F6	64	40
F7	65	41
F8	66	42
F9	67	43
F10	68	44
F11	87	57
F12	88	58
INS (Insert key)	82	52
DEL (Delete key)	83	53
Home	71	47
End	79	4F
Page Up	73	49
Page Down	80	51
Up Arrow	72	48
Down Arrow	80	50

**Table F-1 Scan Codes**

Key	Decimal	Hexadecimal
Right Arrow	77	4D
Left Arrow	75	4B
NUM (Number Lock key)	69	45
/ (divide on Numeric Keypad)	53	35
- (minus on Numeric Keypad)	74	4A
+ (plus on Numeric Keypad)	78	4E
Print Screen	55	37

## SyncToServer

Avstar's timing feature synchronizes the clock on the local workstation with the time set on the server when a user activates show timing. A user can use the Set Clock option from the Tools drop-down menu to manually override the clock synchronization. This feature is turned off by default, but a system administrator can turn it on at any workstation by creating a new registry value in the workstation.



*The synchronized timing feature should be enabled only at those workstations used to time a show. If the SyncToServer registry value is not created and defined at a workstation, then the synchronized timing feature is disabled at that workstation.*

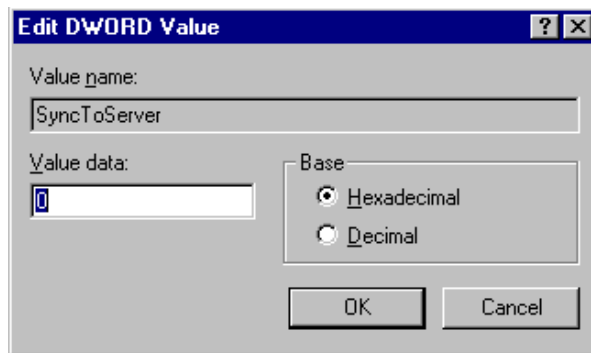


**This feature is not backwards compatible to Avstar NRCS software prior to version 1.2.4. Before enabling the synchronized timing feature by creating the SyncToServer Registry value in any workstation, the software for both Avstar Workstation(s) and Avstar Server(s) must be upgraded. Avstar NRCS will issue the following message if the server does not support the sync-to-server feature: Timing synchronization failed. Not supported on this server.**



To enable the synchronized timing feature, do the following:

1. Open Registry Editor.
2. Navigate to the `Environment` key, and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu will appear. See Figure F-1 “Registry Editor Window” on page F-4.
4. Select the `DWORD Value` option to create and define a new Registry value of type `DWORD` in the Registry Editor.
5. Type the name of the new value: `SyncToServer`.
6. Set the Value data option by doing the following:
  - a. Right-click on the `SyncToServer` value.
  - b. Choose `Modify` from the pop-up menu.
  - c. The `Edit DWORD Value` dialog box will appear.



- d. Set the Value data. Type `0` (zero) to disable the synchronized timing feature, or `1` to enable it.
  - e. Click `OK` to save the setting and close the dialog box.
7. Close the Registry Editor window.

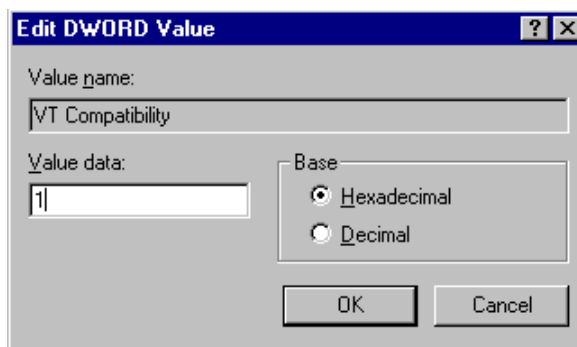
## VT Compatibility

System administrators can set a limit for text in story form fields. For this feature to work, a Registry value defined as `VT Compatibility` must be added in the `Environment` key of the Registry at each workstation. If a registry value is not found in the `Environment` key of the Registry, then the default behavior—no text limit—is observed at the workstation.

To create and define this value, do the following:

1. Open Registry Editor.
2. Navigate to the `Environment` key, and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu will appear. See Figure F-1 “Registry Editor Window” on page F-4.
4. Select the `DWORD Value` option to create and define a new Registry value of type `DWORD` in the Registry Editor.
5. Type the name of the new value: `VT Compatibility`
6. Press Enter.
7. Set the Value data option by doing the following:
  - a. Right-click on `VT Compatibility`
  - b. Choose `Modify` from the pop-up menu.

The Edit DWORD Value dialog box will appear.



- c. Type 1 in the Value data field.
  - d. Click OK to save the setting and close the dialog box.
8. Close the Registry Editor window.

After the field property is changed, it is recommended the user sign off and back on so the new text limit can take effect.



*Changing the field width from the queue's right-click pop-up menu has no effect on the limit of the text entered. That limit is still based on the field property set for the form assigned to the queue. Consequently, when editing text in the Queue panel, the field's properties for the form assigned to the queue are in effect. When editing text in the story form, the field properties of the story form are in effect. These properties—for the story and queue forms—may be different from one another.*

## Delete\_Notify

When a deleted queue record was received by the rxnet utility program (from a txnet utility program on another news system) into an update queue, and the queue had a notify group associated with it, an alert was sent to the notify group with the queue name and title (slug) of the deleted entry (story). This is no different than the notification for new or modified stories entering the queue. This resulted in users

wasting time searching for deleted stories—they were no longer in the queue.

Action servers and txnet links behaved the same way as the rxnet utility program. In Avstar NRCS version 1.3, these programs were changed to be more consistent with user initiated operation—that is, the utility programs (servers) will no longer send out notifications when deleting queue entries.

However, as a precaution, for sites which may want such notification to occur, an environment variable can be set to restore the former operation. The environment variable is defined in the workstation's Registry as `delete_notify`. It must be added in the `Environment` key of the Registry and should be set to a non-zero value.

# APPENDIX G

## *Managing Traits at Console*

Chapter 4, “Users,” explains how the system administrator can access and change the various user traits associated with each user’s account from any Avstar Workstation. Chapter 5, “Stories, Queues, and Directories,” explains how to manage database traits from a workstation. Chapter 6, “Groups,” explains how to create groups and apply the system’s group-related features to customize the system’s security and usage from a workstation. This Appendix shows how some of the information also can be viewed and modified from the Avstar console.

- Viewing User Traits from the Console
- Modifying User Traits from the Console
- User Traits Console Command Summary
- Managing Database Traits from the Console
- Changing Database Traits from the Console
- Database Traits Console Command Summary

## Viewing User Traits from the Console

From the Avstar console, use the `list u-v` command to get user trait information. The console will display a verbose list of user accounts.

To get information about a single user, follow the command with the User ID of the specific user. For instance, if you wanted to access the user account, `danielmi`, the command would look like this:

```
list u-v danielmi
```

The result of the command will look something like this:

```

user      rr      kb  su  m  SOEKCVTH  sc  queues
danielmi  180    0   n   i  -OEKCVTH  sc  dest: PEOPLE.D.DANIELMI.NOTES
                                     home: PEOPLE.D.DANIELMI
                                     mail: PEOPLE.D.DANIELMI.MAIL

AVSTAR-A:
```

The first line of the display lists the traits; the other lines list specific values for the traits.

Here is the interpretation of the sample display above:

- User `danielmi` has a read rate (`rr`) of 180.
- His keyboard preference (`kb`) is 0.
- He is a superuser (`su`). The `n` means “news superuser.” A minus (-) would appear if the user is not a superuser.
- His edit mode (`m`) is insert.
- The traits indicated by `SOEKCVTH` and `sc` are explained in Table G-1 “User Traits Summary” on page G-11. See also “list u” on page A-25.
- His destination queue is `PEOPLE.D.DANIELMI.NOTES`.
- His home directory is `PEOPLE.D.DANIELMI`.
- His mail queue is `PEOPLE.D.DANIELMI.MAIL`.

“User Traits Console Command Summary” on page G-11 shows user traits, their Console abbreviations, and detailed information about them.

## Modifying User Traits from the Console

You must be a superuser or user manager (`umanager`) to change user traits. For an explanation of the superuser privileges, see Chapter 3, *Getting Started*. For an explanation of the `umanager` account and privileges, see “Creating a User Manager Account” on page 4-35.

Use the `utraits` command (which requires superuser privileges) to modify a user’s traits from the console. The syntax of the `utraits` command is:

```
utraits userid {trait new-value [trait new-value
...]} {+|-flag}
```

For instance, to set the read rate for a user named Daniel Mitchell, whose User ID is `danielmi`, the command would look something like this:

```
AVSTAR-A# utraits danielmi readrate 195
1 user records modified
AVSTAR-A#
```

To give him superuser privileges:

```
AVSTAR-A# utraits danielmi su n
1 user records modified
AVSTAR-A#
```

To take superuser privileges away from him:

```
AVSTAR-A# utraits danielmi su -
1 user records modified
AVSTAR-A#
```

The blacklist trait is a flag; that is, it is either on or off. You grant flag traits with a plus sign; you take them away with a minus sign. To blacklist the user:

```
AVSTAR-A# utraits danielmi +b
1 user records modified
AVSTAR-A#
```

To remove him from blacklist status:

```
AVSTAR-A# utraits danielmi -b
1 user records modified
AVSTAR-A#
```

You can change more than one trait at a time. For instance, to give Mitchell keyboard 3 and make `SHOW.RUNDOWN` his destination queue, type:

```
AVSTAR-A# utraits danielmi key 3 dest show.rundown
1 user record modified
AVSTAR-A#
```

## Users' Passwords

To change Smith's password to `changeme`, do the following:

1. Become a superuser.
2. Type the `utraits` command using the following format:  
`utraits <username> password <newpassword>`

For instance:

```
AVSTAR-A# utraits smith password changeme
1 user records modified
AVSTAR-A#
```



Since everything you type at the console is recorded in console history, consider using the `force` command to require the user to change his password the next time he logs in. This will help prevent someone from using passwords obtained from the console history.



For users who do not change their passwords, no matter how many times you remind them, use the `force` console command to require them to change passwords at their next login. The `force` console command has this format:

```
force <user or group names>
```

The most common way to use this command is to require a particular user or users to change their password the next time they log in.

For instance, suppose that you have been unable to convince Mitchell and Schofield to change their passwords. As a last resort, to require that they do so the next time they log in, become a superuser and type the following:

```
AVSTAR-A# force mitchell schofield
```

A message similar to the following appears:

```
Users who will be forced to make password changes on next login:
-----
mitchell  schofield
2 users qualified out of a domain of 2 users, and were updated.
```

The `force` command always tells you who is going to be required to change their passwords. The example above reports that it will make both Mitchell and Schofield change their passwords.

If you find that a user does not have a password, use the `force` command to require the user to select a password the next time he or she logs in.

The following example uses the `force` console command to make Weisman select a new password at the next login. To do this, type the following:

```
AVSTAR-A# force weisman
```

A screen similar to the following appears:

```
Users who will be forced to make password changes on next login:
-----
weisman
1 users qualified out of a domain of 1 users, and were updated.
```

You can use the `force` console command to require that anyone who has not changed passwords since a certain date or within a certain date range do so. You can also use this command to force a particular group of users or all your users to change their passwords.

To do these things, use the `force` console command with this format:

```
force [-q] [passchg<date or date range>] <user or group names>
```

Normally, the `force` command tells you which users must change their passwords the next time they log in. If you would rather not see this display, suppress it with the `-q` option.

You may want to use the `force` command to require all users who last changed their passwords prior to a certain date to change their passwords the next time they log in. You can do this by specifying a date in the command, as shown in the format description above.

The `force` command recognizes dates in the same way the `list u` console command does; you can specify relative dates, absolute dates, and date ranges. This command applies only to people who last changed their passwords within the date parameter you set.

If you specify a value for the date parameter, the `force` console command works only on those users who are among those you specify, and whose last password change falls within the criteria set by the date parameter.

For instance, the following command affects only those members of the “producers” group who last changed their password before July 5, 2000:

```
AVSTAR-A# force passchg<05JUL2000 producers
```

To force all users to change their passwords, put a hyphen (-) in place of any user or group names. This is especially useful in combination with a date parameter. For instance, to force all users who last changed their passwords more than 90 days ago, become a superuser and type the following:

```
AVSTAR-A# force passchg<90. -
```

A message similar to the following appears:

```
Users who will be forced to make password changes on next login:
-----
erickson   mccormack   arlin
3 users qualified out of a domain of 62 users, and were updated.
```

### Listing Users Who Do Not Have Passwords

To check for users who do not have a password from the Avstar console, type:

```
list password= u
```



*Ensure that you include a space between the = and the u.*

This lists every user who does not have a password.

```
AVSTAR-A: list password= u
user                rr  kb  su  mode  destination
weisman             0  0  -   o
```

Here, there is one user, weisman, who does not have a password.

To find out who has not changed their password within a specified period of time, use this form of `list u`:

```
list passchg<date> u [<domain>]
```

For instance, to list users who have not changed their password in the last 90 days, enter `list passchg` followed by `<` and the number of days you want to specify (90 in this case) and a period (`.`). Ending the number with a period indicates the value is in days; no period indicates hours. There must be no spaces between `passchg`, the `<`, and the number of days.

For information on users who have not changed their password within the last 90 days, type:

```
list passchg<90. u
```

A screen similar to the following appears:

User	DEV	Date Created	Last Login	Last Password
levy	427	02JAN2000 10:50am	24JUL2000 9:03am	06JAN2000 9:50am

As you can see in the previous example, this produces a listing with:

- Name of user
- Device where the user logged in last
- When user became a user
- Date of the last login
- Date when password was last changed

In the previous example, only user Levy has not changed passwords within the last 90 days. So, use the `send` console command to send a message reminding Levy to change passwords:

```
AVSTAR-A: send levy Please change your password.
```

A message similar to the following appears:

```
message sent to levy
```

Use the less than (<) and greater than (>) operators to specify whether you want to list people who last changed their password *before* (<) or *after* (>) a certain date. Here are a couple examples:

```
list passchg<05JUL2000 u
```

produces a list of all users who last changed their password before July 5, 2000

```
list passchg>10. u
```

produces a list of users who have changed their password after 10 days ago (that is, within the last 10 days).

You can also use `list u` to list people who last changed their passwords sometime between two dates, such as between June 15, 2000

and July 1, 2000. You can even use this command to check a single user or a group of users.

To do these things, use this format of the `list` command:

```
list passchg>date1<date2 u [<user or group names>]
```



*The `date1` and `date2` parameters are not surrounded with greater than and less than characters here as is customary for parameters. These characters are also used in the command alone, and this could cause confusion.*

You must follow `passchg` with a date. This date may be a relative date, an absolute date, or a date range. Also, there must be no spaces between `passchg` and the date or date range, or the `list u` command does not work correctly.

A *relative date* is one that you specify as some time prior to the present date, as in `list passchg<90. u`. In the previous example, we used a relative date to find out which users had last changed their password prior to 90 days ago. Remember, ending the number with a period (.) indicates that the value is in days; no period indicates hours.

An *absolute date* specifies an actual calendar date. In the following example, we use an absolute date to find out which users last changed their passwords before August 5, 2000, by typing the following:

```
list passchg<05AUG2000 u
```

Information similar to the following appears:

User	DEV	Date Created	Last Login	Last Password
-----	-----	-----	-----	-----
mitchell1495		02JAN2000 10:55am	24JUL2000 9:00am	07JAN2000 9:50am

You must specify absolute dates in DDMMYY format. You must enter the days in double-digit format, meaning you must add a leading zero to single digit days, such as 05. Also, you must enter months as they are defined in the Words dictionary.

You can also specify a date range. This way, you can list users who changed their passwords sometime between two specific dates

(date1 and date 2). For instance, to see if anyone changed their password after August 1, 2000, and before August 15, 2000, type the following:

```
list passchg>01AUG2000<15AUG2000 u
```

Information similar to the following appears:

User	DEV	Date Created	Last Login	Last Password
loyd	433	02JAN2000 11:50am	25AUG2000 9:01am	05AUG2000 9:05am

You can also use `list u` to check on particular users or groups. To do this, follow the `u` in the command with user names or groups you want to check.

For instance, suppose that of all your users, only Mitchell and Schofield regularly forget to change their passwords. To see if they have not changed their passwords in the last 90 days, type the following:

```
list passchg<90. u mitchell schofield
```

Information similar to the following appears:

User	DEV	Date Created	Last Login	Last Password
mitchell1495		02JAN2000 10:55am	24JUL2000 9:00am	07JAN2000 9:50am

This causes `list u` to report only on Mitchell and Schofield, and only if they have not changed their passwords within the specified period of time. In the example above, Mitchell has not changed his password in the last 90 days, but Schofield has.

If you specify a group, such as `producers`, `list u` checks members of that group and then reports those that have not changed their passwords in the specified period of time.

Once you have set a policy on how often people must change their passwords, use `list u` regularly to ensure that no one forgets to do this within the prescribed period of time. If one or more users do not

change their passwords often enough, use the `force` command to force them to do so. See “Users’ Passwords” on page G-4 for more information.

## User Traits Console Command Summary

Table G-1 is a summary of Avstar user traits. The first column shows the trait name as it appears in the Modify User Account dialog box, which is explained in detail in Chapter 4. See “Modify User Account Dialog Box” on page 4-4 for more information.. The second column lists actual commands for assigning or removing each trait. The third column contains an explanation of the trait and example of utraits console command lines.

**Table G-1**      **User Traits Summary**

Name in Modify User Account dialog box	Utrait Console Command	Definition / Example
Superuser	<code>su n   su -</code>	The user has superuser privileges.
Blacklisted	<code>+b   -b</code>	The user cannot log in to the Avstar system.
Simplified	<code>+s   -s</code>	The user has limitations, as defined by the Simplified User Settings.  Examples:  <pre>utraits palmer su n utraits palmer su - utraits palmer +b utraits palmer +s</pre>
Insert	<code>editmode i</code>	Everything the user enters in a story is inserted at the current cursor position, moving the following text over.  Example:  <pre>utraits palmer editmode i</pre>

**Table G-1 User Traits Summary (Continued)**

Name in Modify User Account dialog box	Utrait Console Command	Definition / Example
Overwrite	<code>editmode o</code>	<p>Everything the user enters in a story overwrites the character under the cursor. This is the default.</p> <p>Example:</p> <pre><b>utraits hansen editmode o</b></pre>
Home	<code>home</code>	<p>Sets the user's home directory, which usually contains the user's Mail and Notes queues.</p> <p>Example:</p> <pre><b>utraits loyd home people.1.loyd</b></pre>
Destination	<code>destination</code>	<p>Specifies the user's destination, which is usually a queue he or she uses frequently, such as the Notes queue.</p> <p>Example:</p> <pre><b>utraits loyd destination people.1.loyd.notes</b></pre>
Mail	<code>mail</code>	<p>Specifies the user's Mail queue, where any mail addressed to that user is placed.</p> <p>Example:</p> <pre><b>utraits loyd mail people.loyd.mail</b></pre>
Read Rate	<code>readrate</code>	<p>The user's spoken reading rate in words per minute.</p> <p>The system uses the readrate of the designated presenter to determine the audio (air) time of a story.</p> <p>Example:</p> <pre><b>utraits vandenberg readrate 180</b></pre>
User Name	<code>realname</code>	<p>The user's real name (as contrasted with his or her account's User ID name). For instance, John Chapman may have a User ID of <code>jchapman</code>; his real name is John Chapman.</p>



**Table G-1 User Traits Summary (Continued)**

<b>Name in Modify User Account dialog box</b>	<b>Utrait Console Command</b>	<b>Definition / Example</b>
Video Browse	+vb   -vb	Specifies user can search the video server. Example: <b>utraits palmer +vb</b>
Broadcast Control	+td   -td	Enables the user to employ the technical director's workstation for broadcast control. Example: <b>utraits miller +td</b>
Connect Services	+C   -C	Allows user to connect to any service defined in the system. Example: <b>utraits hinkle +C</b>
Toolbars	+cs   -cs	Allows users to create custom toolbars. Example: <b>utraits carver +cs</b>
Color Highlights	+cc   -cc	Allows users to configure custom status colors. Example: <b>utraits bagley +cc</b>
Highlight Read Stories	+hr   -hr	Specifies that unread stories in the queue are highlighted on the user's screen. Example: <b>utraits foley +hr</b>

**Table G-1 User Traits Summary (Continued)**

Name in Modify User Account dialog box	Utrait Console Command	Definition / Example
Reorder Stories	+o   -o	Specifies that user can create and remove stories from a folder or queue. Example: <code>utraits lumpkin +o</code>
Create/Kill Folders/Queues	+er   -er	Specifies that user can create and remove entire folders and queues. Example: <code>utraits delany +er</code>
Kill All Stories	+ka   -ka	Specifies that user can select an entire queue and move stories in it to the Dead queue. Example: <code>utraits wright +ka</code>
Password button	password	Provides capability to change the user's password. Example: <code>utraits jordan pass changeme</code>
Force Change check box	force	Specifies that user will be forced to change his/her password the next time he or she logs in. This command line does not require <code>utraits</code> typed in front of it, as shown below. Example: <code>force jordan</code>
User Preferences button; Sessions tab; Keyboard field	key ###	Assigns a default keyboard (set of macros) to the user account. Example: <code>utraits vanderberg key 048</code>

## Managing Database Traits from the Console

### Getting Basic Information

To obtain information about a particular queue or directory from the Avstar console, type `list d`, followed by the name of the directory or queue.

For instance, to obtain information about the `WIRES.ALL` queue, type:  
**`list d wires.all`**

Information similar to the following appears:

```
SRPlo-LIsUGQsXWFi sortfield purge dis mbox directory
-R-----I---Q-X---  TITLE P4.0 D1          - WIRES.ALL
```

A trait listing begins with a header line containing the letters `SRPlo-LIsUGQsXWFi`; each letter in the header line represents a particular database trait. That is, in the previous example, the second letter in the header line (an `R`) stands for the database trait Read-only.

When one of these traits is on, the letter representing that trait appears in the second line. For instance, the `R` in the second line of the example indicates the Read-only trait is on:

```
SRPlo-LIsUGQsXWFi sortfield purge dis mbox directory
-R-----I---Q-X---  TITLE P4.0 D1          - WIRES.ALL
```

A hyphen in the second line indicates that trait identified above it is off. For instance, the first trait in the header, an `S`, represents a sequentially ordered directory or queue. Because the second line has a hyphen below the Sequential trait indicator (the `S`) means that `WIRES.ALL` is alphabetically, rather than sequentially, ordered.

Another trait indicated as “on” for `WIRES.ALL` is the one represented by the letter `I`. This trait is Inverted, which means this queue displays the most recent stories at the top.

```
SRPlo-LISUGQSWFi sortfield purge dis mbox directory
-R-----I---Q-X---  TITLE P4.0 D1          - WIRES.ALL
```

Database traits are all listed and explained in Table 5-1.

## Getting Detailed Information

To obtain a more detailed information about a directory or queue, add the verbose command, such as `list d-v` followed by the queue or directory name.

For instance, to list information about the queue called `RUNDOWNS.5PM`, type:

```
list d-v rundowns.5pm
```

A screen similar to the following appears:

```
SRPlo-LISUGQSWFi sortfield purge ap,al, as dis mbox
RUNDOWNS.5PM:
QSRP-----U-Q----- TITLE   P0      A000,000,000 D1 220
      rg=castread wg=castwrite ng=-
queue form=MCS-RUNDOWN story form=MCS-RUNDOWN
```

The first letter (a `Q`) in the line under the queue name indicates that `RUNDOWNS.5PM` is a queue.

The `ap` near the end of the header line stands for Abstract Printer. This trait indicates which printer your system uses to automatically print abstracts for this queue.

The `al` that follows `ap` stands for Abstract Lines. This trait controls whether or not abstracting is done, and, if it is done, how many lines of each story are printed.

## Changing Database Traits from the Console

To change a database trait from the Avstar console, you must use the `dbtraits` command. The general format is:

```
dbtraits pathname [only][option value][+mode][...]
[-mode][...]
```

Database traits come in two types: options and modes.

- Options accept a range of values, such as setting 18 hours for a queue's purge interval.

```
dbtraits rundowns.5pm purge 18
```

- Modes are traits that are either assigned or not—that is, they are either turned on or off. A trait preceded by a `+` turns a mode on. A trait preceded by a `-` turns it off. Here's an example:

```
dbtraits rundowns.5pm +p
```

You can change several traits at the same time. For instance, the following command changes the queue to read-only status and also assigns the wires form to it:

```
dbtraits wires +r form wires
```

### Changing a Parent Directory Only

Normally, when you change a directory's traits at the console, `dbtraits` also applies your changes to any subdirectories or queues in that folder. You can restrict your changes to the parent directory by following the directory name with the word `only`.



*This is directly opposite to how traits are assigned using the Avstar Workstation. See "Viewing Database Traits" on page 5-13 for more information.*

For instance, to turn on the Scripts directory's Sequential trait (`+s`) without also turning it on for any of the Scripts subdirectories or queues, type:

```
dbtraits scripts only +s
```

## Database Traits Console Command Summary

Table G-2 is a summary of Avstar database traits, (SRPlo-LIsUGQsXWFi) as seen and used in connection with the `dbtraits` command at the Avstar console.

**Table G-2 Database Traits Summary**

Name	Location in Display SRPlo-LIsUGQsXWFi	Mode/Option Keyword	Definition / Example
Sequential	S-----	+s   -s	<p>Lists directories or queues in the order in which they were created. (The default is alphabetical.)</p> <p>To order the <code>RUNDOWN.5PM</code> sequentially:  <code>dbtraits rundown.5pm +s</code></p>
Read Access	-R-----	+r   -r	<p>Indicates whether or not stories in the queue are in read-only mode.</p> <p>To set the <code>SHOW.5PM.SCRIPTS</code> queue to read-only mode, type:  <code>dbtraits show.5pm.scripts +r</code></p> <p>When this mode is turned off, users opening stories have them in edit-lock mode. This is the logical setting for any queue in which people will be changing stories.</p> <p>Turn Read Access on for queues in which people are likely to read but not change the stories.</p>
Printable	--P-----	+p   -p	<p>Indicates whether you can use the print command on all stories in the queue with a single command.</p> <p>To enable users to print all stories in the <code>SHOW.5PM.SCRIPTS</code> queue with a single command:  <code>dbtraits show.5pm.scripts +p</code></p> <p>This trait does not interfere with your use of the <code>print story</code> or <code>print script</code> commands on individual stories in the queue.</p>

**Table G-2 Database Traits Summary (Continued)**

Name	Location in Display SRPlo-LIsUGQsXWFi	Mode/Option Keyword	Definition / Example
Queue Being Ordered	---lo-----	-o	<p>This is an indicator, rather than database traits. It indicates the queue's order status.</p> <p><b>l</b> indicates that the queue is currently order locked to prevent more than one user from reordering stories in a queue at the same time.</p> <p>To find out who is ordering a queue, read the Busy error message you get when you try to order the queue. If no one is actually ordering the queue, then it has an invalid order lock on it.</p> <p><b>o</b> indicates the queue was once sorted, but has since been ordered. When a sorted queue is ordered, the system stops sorting stories as they enter the queue. To indicate that a sorted queue has been ordered, the system replaces <code>sorted</code> on the screen with <code>order</code>.</p> <p>The <code>dbtrait</code> command <code>-o</code> can be used to remove the ordered attribute (indicator) and make the queue resume sorting from the console. There is no <code>+o</code> to apply the ordered attribute. See "Starting the Queue Sort Function from the Console" on page G-26 or "Starting the Queue Sort Function" on page 5-33 for more information.</p> <p>To find out who last ordered the  <code>SHOW .5PM.SCRIPTS queue:</code>  <code>list d-o show.5pm.scripts</code></p>
Locked Queue	-----L-----		<p>Not a database trait; indicates a user has locked the queue. Only a superuser or someone who knows the correct key for the lock can remove it.</p> <p>To find out who last locked the  <code>SHOW .5PM.SCRIPTS queue:</code>  <code>list d-u show.5pm.scripts</code></p>

**Table G-2 Database Traits Summary (Continued)**

Name	Location in Display SRPlo-LIsUGQsXWFi	Mode/Option Keyword	Definition / Example
Inverted	-----I-----	+i   -i	<p>Indicates whether or not stories in a directory or queue will be displayed with the most recent one at the top.</p> <p>To display stories in the <code>WIRES.ALL</code> queue, with the most recent one at the top:  <b>dbtraits wires.all +i</b></p>
Sorted	-----s-----	+so   -so	<p>Indicates whether or not the stories in a queue will be sorted by a form field you choose (usually the title field).</p> <p>To turn on the sorted trait for the <code>SHOW.5PM.SCRIPTS</code> queue:  <b>dbtraits show.5pm.scripts +so</b></p>
Update	-----U-----	+u   -u	<p>Indicates whether or not stories in a queue will be replaced as new versions are moved or copied to it.</p> <p>To indicate that stories in <code>SHOW.5PM.SCRIPTS</code> should be replaced as new versions are moved or copied to it:  <b>dbtraits show.5pm.scripts +u</b></p> <p>The update trait does not affect stories that are restored from tape. If you restore a story to a queue that already contains a version of that story, you will have two versions of the same story, even if the queue's update trait is on.</p>
General	-----G-----	+g   -g	<p>Indicates whether or not stories in a queue should retain their security restrictions when they are moved to another queue.</p> <p>For instance, access to stories in the <code>Dead</code> queue would normally be unrestricted. To prevent people from opening restricted stories once they are moved to the <code>Dead</code> queue:  <b>dbtraits dead +g</b></p>



**Table G-2 Database Traits Summary (Continued)**

Name	Location in Display SRPlo-LIsUGQsXWFi	Mode/Option Keyword	Definition / Example
Queue Operations Allowed	-----Q----	+q   -q	<p>Indicates whether or not the kill, move, or duplicate operations can be performed against an entire queue.</p> <p>Setting the trait “off” for a particular queue does not affect the ability of people to kill, move, or duplicate individual stories in the queue, as long as they have appropriate permissions.</p> <p>To allow users to kill, move, or duplicate the entire <code>SHOW.5PM.SCRIPTS</code> queue:  <b>dbtraits show.5pm.scripts +q</b></p>
Save Version	-----S----	save - n o a	<p>Determines how many old story versions are retained in each queue.</p> <p>To save only the last version of the People directory (this is the default) use:  <b>dbtraits people save -</b></p> <p>To save no version use:  <b>dbtraits people save n</b></p> <p>To save only the original version use:  <b>dbtraits people save o</b></p> <p>To save all versions use:  <b>dbtraits people save a</b></p> <p>To access a list of saved stories use:  <b>display all</b> (VT interface only)</p> <p>To open an old version of a saved story, use:  <b>read old</b> or <b>get old</b></p> <p>To read all old versions of a story, use:  <b>read all</b></p> <p>To edit all old versions of a story, use:  <b>get all</b></p>

**Table G-2 Database Traits Summary (Continued)**

Name	Location in Display SRPlo-LIsUGQsXWFi	Mode/Option Keyword	Definition / Example
Skip	-----X--	+x   -x	<p>Indicates whether or not a directory or queue is left out of database backups.</p> <p>To leave the Dead queue out of the database backup procedure:  <b>dbtraits dead +x</b></p>
Watch Append	-----W--	+w   -w	<p>Assigned to queues that receive data from wire services. It allows the queue to monitor for new stories sent by the wire service, appends them to the wire queue, and immediately displays them to users who have that wire queue open. While this trait can be applied to any queue in Avstar NRCS, it is crucial that it be assigned to queues receiving data from wire services. For instance, the command to assign Watch Append to the WIRES.ALL queue is:  <b>dbtraits wires.all +w</b></p>
Forms Allowed	-----F-	+f   -f	<p>Must be assigned to all queues in the forms directory. The forms will not work without this database trait applied. Additionally, this trait can be assigned to any queue in the database, but is usually only assigned to other queues that receive stories from other systems via rx/tx and then build forms for those stories, as needed. For instance, the command to assign the forms allowed trait to the SYSTEM.FORMS.R queue is:  <b>dbtraits system.forms.r +f</b></p>
Index	-----i	+index   -index	<p>Assigned to queues and directories that you want to be indexed by the Fast Text Search (FTS) server. Allows for quicker searching of the queue or directory. For instance, the command to assign this trait to the WIRES.ALL queue is:  <b>dbtraits wires.all +index</b></p>

**Table G-2 Database Traits Summary (Continued)**

<b>Name</b>	<b>Location in Display</b>	<b>Mode/Option</b>	<b>Definition / Example</b>
Sortfield	sortfield	sortfield	Indicates which form field a sorted queue is sorted by. See "Sortfield" on page G-25 for more information.
Purge Interval		purge	Indicates the "age" stories in a queue can reach before they are purged. See "Purge Interval" on page G-27 for more information.
AbstractPrinting		ap as al	Indicates whether or not a queue is an abstract printing queue. See "Abstract Printer" on page G-29 for more information.
Display Lines/Refresh		dis / + -refresh	Indicates how many lines of each story in the queue are displayed. See "The dis Column" on page G-31 for more information.
Queue Form		qform	<p>The form used to display information in the Queue panel. Fields selected should be a sub-set of fields used in the story form. A field included in the queue form that does not actually exist in the story form cannot be written to in the Queue panel.</p> <p>In this example the rundown form is assigned as the queue form to the SHOWS.6PM.RUN-DOWN queue:</p> <pre><b>dbtraits show.6pm.rundown qform rundown</b></pre>
Story Form		sform	<p>Indicates the form assigned to be used in the Story Form panel (of the Story panel) to display form fields.</p> <p>In this example the futures form is assigned as the story form to the ASSIGNMENTS.TODAY queue:</p> <pre><b>dbtraits assignments.today sform future</b></pre>

**Table G-2 Database Traits Summary (Continued)**

Name	Location in Display SRPlo-LIsUGQsXWFi	Mode/Option Keyword	Definition / Example
Change Form		cform	<p>Changes the story form assignment for previously existing stories within a queue, and restamps them with a new form name. The following example would change the way stories in the 6 o'clock show would appear, reassigning them to be displayed using a newly designed rundown form.</p> <p>Example:</p> <pre>dbtraits show.6pm.rundown sform rundown-new cform</pre> <p>For this to take effect, you need to log out and log back in again.</p>
Strip Form		stripform	<p>Removes embedded form display information from stories. Forms allowed queues stamp the look of the story form into the story. Assigning a different story form to one of these queues and running <code>changeform</code> on the queue would not affect the look of stories with embedded forms. You would need to strip the embedded "look" out of the story so it would then appear using the form assigned to that queue.</p>
Mailbox	mbox	mail	<p>Indicates mailbox assigned to the queue. See "Mailbox" on page G-31 for more information.</p>
Groups	wg= rg= ng=	wg   rg   ng	<p>Assigns a write, read, or notify group to queue or directory. Here are a few examples:</p> <pre>dbtraits show.5pm rg=castread dbtraits show.5pm wg=producers</pre> <p>See "Managing Group Traits at the Console" on page G-34 for more information. See "Groups" on page 6-1 for more information.</p>

## Sortfield

The format of the `sortfield` information is:

```
SRPlo-LIsUG-QSXWF sortfield
```

```
Q----- TITLE
```

This trait shows by which form field a sorted queue is sorted. A letter representing the form field appears in this position in the trait listing. A hyphen (-) here means that no sort field has been set. The system automatically uses the title field as the sort field.

This trait also determines which form field the computer searches during fast finds. In addition, the cursor is placed on this form field when a user displays stories in a queue.

Do not confuse this trait with the sorted queue trait, which determines whether the queue is sorted at all.

### Changing a Queue's Sort Field

Perform this procedure at a time of low system usage, because the `dbsort` command can impair system performance while it is running.

To change a queue's sort field from the console, do the following:

1. Make sure no users are in the queue.

Changing a queue's sort field does not affect what is seen by users already in the queue. If they try to use a displayed out-of-date sort field, it could cause confusion.

2. Use `dbtraits` to set the queue's sort field.

```
dbtraits <queue name> sortfield <field name>
```

If the queue is not already sorted, include `+so` at the end of the command to turn on its sorted trait.

For instance:

```
dbtraits rundowns.5pm sortfield page-number +so
```

Turn on the sorted option (`+so`) before allowing users to re-enter the queue.

3. Type `dbsort` to make stories in the queue adopt the new sort field.

```
dbsort -v <queue name>
```

4. To view the sortfield, type `list q`.
5. To make stories in the `RUNDOWNS . 5PM` queue adopt the new sort field, type:

```
dbsort -v rundowns.5pm
```

A display similar to the following appears:

```
verifying: RUNDOWNS.5PM
quick index [noon avstar harris] will be [xxx      ] for 9283
...
```

6. Type `dbsort` again to sort stories.

```
dbsort rundowns.5pm
```

Users can now re-enter the queue, and stories in it are sorted by page number. When users do a fast search, your system searches on whatever form field has been set as the queue's sort field. Unless you have changed the sort field, this is almost always the title field.

When the user is in a queue looking at a number of stories, if the sort field is on line two of the form, the cursor is placed on the sort field instead of in the title field. If neither the title field nor the sort field appear on line two of the form, the cursor is placed in column one of line two of the form.

### Starting the Queue Sort Function from the Console

You can use the Avstar console to restore a ordered queue to its original sorted state and restart sorting in one of two ways:

- Use the `dbsort` command
- Use the `dbtraits -o` command

To restart sorting at a queue using the `dbsort` command, do the following:

1. Type the `dbsort` command and the queue name you want to sort. For instance:

```
dbsort rundowns.5pm
```

2. Press Enter. Something similar to the following will appear:

```
Sorting: <RUNDOWNS.5PM>
```



*If the queue has a write group assigned to it, you must be a console superuser to use `dbsort`. Otherwise, you will see something like this:*

```
RUNDOWNS.5PM Write group producers NOT sorted.
```

To restart sorting at a queue using the `dbtraits -o` command, do the following:

1. Type the `dbtraits` command, the queue name you want to sort, and `-o`. For instance:

```
dbtraits rundowns.5pm -o
```

2. Press Enter.

Either of these commands turn off the ordered attribute, allowing the queue to resume sorting, as indicated by the sort trait.

## Purge Interval

The format of the `purge` information is:

```
purge ap, al, as dis temp mbox
```

```
P3.0 A000,000,000 D7 T0 -
```

The `purge` column in the `list d` output displays the directory or queue's purge interval, preceded by `P`. The purge interval determines how old stories in a queue can get before they are purged. Every hour, your system removes any stories older than their queue's purge interval and places these stories in the Dead queue. This process frees up space in your database for new stories.

You can set a purge interval as days or hours, or a combination of both. You distinguish between hours and days by using a decimal point.

To enter hours, type the number of hours, such as 24.

To enter the purge interval in days and hours, type the number of days, a period, and the number of hours, such as 2.12.

To set the purge interval in days, type the number of days followed by a period (for instance, 90.). A purge interval of zero means the queue is never purged.

To set a queue's purge interval, use:

```
dbtraits <queue name> purge <purge interval value>
```

For instance, to give all queues in the Wires directory a purge interval of two days, type:

```
dbtraits wires purge 2.
```

To list all queues in the database with a purge interval, use:

```
list purge=<purge interval> d
```

To list all queues in a directory with a purge interval, use:

```
list purge=<purge interval> d [<directory name>]
```

## Abstract Printing

This group of traits controls whether or not a queue is an abstract printing queue. When a story is added to or modified in an abstract printing queue, the system automatically prints a portion of the story. Using these traits, you can control how much of each story is printed and the print style used to print the story.

Abstract printing is controlled by three traits: abstract printer, abstract lines, and abstract style.



To list a queue's abstract printing traits, type **list d-a** followed by the queue name. For instance, to list the abstract traits of `WIRES.STATE`, type:

```
list d-a wires.state
```

A display similar to the following appears:

```
SRPlo-LISUG-QSX s ap, al, as  dis temp directory
D--P-----Q-- - A001,005,008 D7 T0  WIRES.STATE
```

The three abstract printing traits appear toward the middle of the listing under the headings `ap`, `al`, and `as`.

### Abstract Printer

The abstract printer trait indicates which printer your system uses to automatically print abstracts for this queue. To select a printer, set this trait to the printer number you want to use. (This is the same number you would use to select that printer in a `print` command.) Use this form of `dbtraits`:

```
dbtraits <queue name> ap <printer number>
```

For instance, to set `WIRES.STATE` to print abstracts on printer 1, type:

```
dbtraits wires.state ap 1
```

The format of the abstract printer information is:

```
purge ap, al, as  dis temp mbox
P3.0 A000,000,000 D7 T0  -
```

To list all queues in the database that have a particular abstract printer value, use:

```
list ap=<printer number> d
```

### Abstract Lines

The abstract lines trait controls whether or not abstracting is done and, if it is done, how many lines of each story are printed.

Both `ap` and `al` must be non-zero for abstract printing to be enabled.

To set this trait, use:

```
dbtraits <queue name> al <abstract lines value>
```

For instance, to set `WIRES.STATE` to print 10-line abstracts, type:

```
dbtraits wires.states al 10
```

If you do not want the queue to be an abstract printing queue, set this trait to zero. Otherwise, set this trait to the number of lines you want printed for each story. You can select a number of lines between 1 and 125. A value of 126 prints the entire story in script form. Setting abstract lines to 127 prints the entire story in story form.

Zero is the default value for this trait, which means the queue is not set up to print abstracts.

The format of the `abstract lines` information is:

```
purge ap, al, as dis temp mbox
```

```
P3.0 A000,000,000 D7 T0 -
```

### Abstract Style

The abstract style trait indicates the print style your system uses when it prints an abstract from this queue. To choose a style, set this trait to the number representing the style you want. To list all queues in the database that have a particular abstract style value, use:

```
list as=<style number> d
```

To set this trait, use:

```
dbtraits <queue name> as <abstract style value>
```

For instance, to set `WIRES.STATE` to print abstracts with style 3, type:

```
dbtraits wires.states as 3
```

You can select any style defined in `SYSTEM.STYLES`.

The format of the `abstract style` information is:

```
purge ap, al, as dis temp mbox
```

```
P3.0 A000,000,000 D7 T0 -
```

## Mailbox

If a mailbox is assigned to a queue, the number representing that mailbox appears in the `mbox` column of the `list d` display.

The format of the mailbox information is:

```
purge ap, al, as dis temp mbox
```

```
P3.0 A000,000,000 D7 T0 -
```

To list all queues in the database with a particular mailbox, use:

```
list mail=<mailbox number> d
```

To assign a mailbox to a queue, use:

```
dbtraits <queue name> mail <mailbox number> |
<reserved mailbox name>
```

For instance to assign mailbox 103 to the queue `WIRES.KEYWORDS.HOCKEY`, type:

```
dbtraits wires.keywords.hockey mail 103
```

## The dis Column

The `dis` column of the `list d` output represents two traits in one column: the preview lines trait (formerly known as display lines) and the refresh trait.

The format of the preview lines information at the console is:

```
purge ap, al, as dis temp mbox
```

```
P3.0 A000,000,000 D7 T0 -
```

The preview lines trait applies to lines previewed in the Queue panel of the Avstar Workstation as well as VT terminals. To see an example of preview lines at the workstation, see “Preview Lines” on page 4-17. The value of a queue’s preview lines trait controls how many lines of each story in the queue are displayed. The number in the `dis` column indicates the preview lines setting. You can have a queue display as few as one line of each story or as many as 23 lines.

The second trait, refresh, is also listed under `dis`. Refresh controls whether the system automatically updates your screen if you view a queue someone else is changing. If the number under `dis` begins with a `D`, the queue is not refreshed automatically; if it begins with an `R`, it is refreshed automatically.

### Preview Lines

To change the value of a queue's preview lines trait, use:

```
dbtraits <queue name> dis <number of lines>
```

Because screens can display a maximum of 23 lines of a story at a time, that is the maximum value you can use for this trait. The minimum value is 1.

For instance, to give `SHOW.RUNDOWNS` a one-line display, type:

```
dbtraits show.rundown dis 1
```

`SHOW.RUNDOWNS` now has a one-line-per-story display. The graphic below shows how this may appear on a VT.

SHOW.RUNDOWNS								sorted
PAGE	TAL	SLUG	FX	UTR	TEXT	TRT	BACK	CUNE
11-A	PU	TS:PARKING	vo-wipe---					:00
11-A	PU	TS:OFF SHORE	vo-wipe---					0:31
11-43	RC	TS:MOON WALK	UTR					:15
11-23	RC	TS:JAPANESE DOC	UTR					:15
11-41	PU	Toss Tom						:10

In a one-line display, the first line on the screen is the top line of the form assigned to that queue. The next 22 lines in the queue display the second line of each story's form. Queues that display more than one line per story begin with the top line of each story's form.

Choosing a value for a queue's display lines trait depends on which is more important: displaying as many stories as possible at one time, or showing a large portion of each story.

For instance, in a rundown queue it is important to display as many stories on the screen at a time as possible, so the display lines trait is usually set to one line per story.

In queues where people often browse (such as those in the Wires directory), display a small part of each story. Here is an example of the `WIRES.STATE` queue displaying three stories on a VT screen simultaneously:

```

WIRES.STATE                                     Read-only
SLUG      TAKE  FROM      MOVED      STATUS TIME
AP-DTrain 06-03 0165      0716      APv0nb--nr Fri Jun 3 11:01 WIRE  :59
ap-D Train-
.
  (NEW YORK) Power failure on a Manhattan-bound D train and a
  passenger taken ill on a following train caused backups this
  morning for thousands of B-M-T subway riders in Brooklyn and lower
SLUG      TAKE  FROM      MOVED      STATUS TIME
AP-TRAFFICADVISORY 06-03 0714      APvnyallnr Fri Jun 3 10:44 WIRE  :16
AP-TRAFFICADVISORY 06-03 0072
ap-TRAFFIC ADVISORY
.
  New York City police report there is an overturned
  Tractor-trailer on the Bronx-bound George Washington Bridge at West
SLUG      TAKE  FROM      MOVED      STATUS TIME
AP-ny-LivestockReport 06 7023      APv0ny--nr Fri Jun 3 10:32 WIRE  1:22
AP-ny-LivestockReport 06-03 0260
ap-ny-Livestock Report-
.
  (albany) The following are livestock sales at auction this
  week as reported by the U.S. Department of Agriculture.
  Slaughter cattle: Demand moderate to good. Compared to last week

```

As in the previous example, stories in a queue display are usually separated by a blank line. The only exception is queues that display one line per story, such as rundown queues. Stories in these queues are not separated from each other.

Type the `display` command to temporarily change the number of lines displayed per story for a queue you are viewing. This change applies only to a single user's workstation and lasts only while the queue is being viewed.

## Refresh

To set the queue refresh trait at the console, use:

```
dbtraits <queue name> [+refresh | -refresh]
```

To turn on this trait for a queue, use `+refresh`; to turn it off, use `-refresh`. For instance, to turn on the refresh trait in `RUNDOWNS.5PM`, type:

```
dbtraits rundowns.5pm +refresh
```

Queues with refresh turned on display `R` in the preview lines column of `list d` output.

Use this trait only on important queues, like rundowns, that more than one person may modify simultaneously. This allows everyone working on that rundown to immediately see changes made by others.



*To automatically refresh a queue, your system must spend a long time monitoring workstations where users are viewing that queue. Using the refresh option on too many queues simultaneously greatly increases the amount of work your system has to do and may severely degrade its overall performance.*

## Managing Group Traits at the Console

### Groups

There are three primary group traits used to apply groups to queues and directories in the database: Read, Write, and Notify.

The `list d-v` console command lists each queue's assigned read, write, and notification groups. These groups restrict who can read or write stories in the queue and indicate who is notified when stories change in it. Each group is explained individually in the following sections.

To list group information for a queue at the console, use:

```
list d-g <queue name>
```

To list all queues in the database with a particular group assigned as their read, write, or notification group at the console, use:

```
list rwnng=<group name> d
```

### Read Group

To set a queue's read group, use:

```
dbtraits <queue name> rg <group name>
```

For instance, to assign a read group of producers to the queue `SHOW.5PM`, type:

```
dbtraits show.5pm rg producers
```

To list all queues in the database with a particular group assigned as their read group, use:

```
list rg=<group name> d
```

### Write Group

To set a queue's write group, use:

```
dbtraits <queue name> wg <group name>
```

For instance, to assign a write group of producers to the queue `SHOW.5PM`, type:

```
dbtraits show.5pm wg producers
```

To list all queues in the database with a particular group assigned as their write group, use:

```
list wg=<group name> d
```

To list all queues in the database with a particular group assigned as their read or write group, use:

```
list rwg=<group name> d
```

### Notify Group

To set a queue's notification group, use:

```
dbtraits <queue name> ng <group name>
```

For instance, to assign a notification group of producers to the queue `WIRES.WAR`, type:

```
dbtraits wires.war ng producers
```

To list all queues in the database that have a particular group assigned as their notification group, use:

```
list ng=<group name> d
```

## Restricting Access Using Read and Write Limitations

In addition to restricting access to various queues, you can use group access and usage restrictions to hide queues or directories by placing a strict read restriction on them.

For instance, the System directory is usually restricted so that only superusers can write stories there. You can hide this directory so it does not appear in the main directory for normal users. Set its read group to a group with no users, such as `sysop`. (Because system administrators can read everything in the database, they can see the directory.)

To set the System directory's read group to `sysop` at the Avstar console, do the following:

1. Become a console superuser so that the prompt appears like this:

```
AVSTAR-A#
```

2. Type:

```
AVSTAR-A# dbtraits system rg sysop
```

3. To apply changes you made to stories already in the directory, remain a console superuser and type:

```
AVSTAR-A# gtraits changegroup system
```

A message similar to the following appears:

```
SYSTEM.CABLES 5 out of 5 changed
...
SYSTEM.WIRES 4 out of 4 changed
```

Now, only system administrators, logged in with an superuser account, can see the System directory on their screens.

You may also have a queue to which you want people to move or duplicate stories, but that you do not want anyone to read or go to. An example of this would be a confidential suggestions queue to which people would move stories containing suggestions. Once placed in this queue, suggestions would be visible only to those authorized or to superusers.



For instance, to set the read group of the Suggestions queue to mngmt, as a console superuser, type:

```
AVSTAR-A# dbtraits suggestions rg mngmt
```



*All users you want to allow the capability to send suggestions need to have write access to the queue.*

To make changes take effect on stories already in the Suggestions queue, remain a console superuser and type:

```
AVSTAR-A# gtraits changegroup suggestions
```

Unless you are a member of the mngmt group, you cannot see this queue in the Directory panel or open this queue. You can copy or move stories containing suggestions to this queue.

## Removing Directory or Queue Restrictions

To remove a directory or queue's read or write restrictions at the Avstar console, you first must become a console superuser—that is, the prompt should appear with a number sign (#) not a colon (:). To remove a restriction, type **dbtraits** followed by the name of the directory or queue you want to change, then **rg**, **wg**, **ng**, and a dash (-).

For instance, to remove both the read and write restrictions from the Phonetists queue, do the following:

1. As a console superuser, type:

```
AVSTAR # dbtraits phonelists rg - wg -
```

2. To apply these read and write group changes to stories already in the Phonetists queue, as a superuser, type:

```
AVSTAR # gtraits changegroup phonelists
```

Now, anyone can read and write stories in the Phonetists queue.



# Glossary

<b>10Base-T</b>	Low-cost point-to-point 10Mb/sec Ethernet using four unshielded twisted pairs (UTP) of wire (only two pairs are actually used) with RJ-45 connectors.
<b>100Base-T</b>	Low-cost point-to-point 100Mb/sec Ethernet using four UTP (only two pairs are actually used) with RJ-45 connectors.
<b>absolute time</b>	The time assigned to a clip when it was encoded.
<b>account</b>	A level of authorization assigned to individuals using Avstar NRCS. This determines the types of information users can access and the actions they can perform. Account types include user, user manager, superuser, and system administrator.
<b>alias</b>	A code of up to 12 alphanumeric characters. It substitutes individual user names and automates the distribution of a mail story to a group of people.
<b>anchor</b>	<ol style="list-style-type: none"><li>1. The person presenting a newscast on-air to a television audience. Also called a <i>presenter</i>.</li><li>2. The indicator in a Story Text panel that links a script to production information. Also called: <i>grommet</i>, <i>grotch</i>, or <i>marker</i>.</li></ol>

<b>ASCII</b>	American Standard Code for Information Interchange. The standard that governs the recording of characters by a sequence of binary digits, as in a computerized timecode or video editing system.
<b>ASWS</b>	An acronym for Avstar Workstation.
<b>auto backup</b>	A function in Avstar that writes a backup copy of an open story to a user's local disk at specified time intervals.
<b>auto-refresh</b>	A queue attribute that automatically redisplay the queue screen whenever changes are made to the queue.
<b>autoscript</b>	A mode in which the production cue area of a story is automatically displayed if production cues are in the story. If there are no production cues added to a story, the story is displayed unscripted.
<b>backtime</b>	The exact time when a story in a newscast must start in order for the show to remain on schedule. Television newscasts typically use back-time to ensure that the newscast ends precisely as scheduled.
<b>baud</b>	Unit for measuring the rate of the digital data transmission. Usually one baud equals one bit per second.
<b>branch</b>	A subordinate segment of the directory as displayed in the Directory panel.
<b>Broadcast Control System (BCS)</b>	A product that works with Avstar to run a show's production devices, such as character generators, still-stores, and videotape devices.
<b>bulletin</b>	An incoming wire story coded as high-priority by a wire service; it is fed directly into the Avstar "priority" queue. Users are informed of its arrival with both an audio signal and lightning bolt icon in the status portion of the Avstar Workspace.
<b>clip</b>	A segment of digitized source material.

---

<b>connect</b>	To call a service that is either local (such as an archive system) or remote (such as Nexis). In Avstar, users connect to services to access data.
<b>context menu</b>	See <i>pop-up menu</i> .
<b>cue</b>	See <i>production cue</i> .
<b>cume (cumulative) time</b>	The amount of airtime required from the beginning of the show up to a certain point in the show in order for the show to remain on-schedule. It is displayed with each entry in a rundown queue. Cumetime is used by producers when building, ordering, or airing a newscast.
<b>DAT</b>	A digital audio recording format that uses 3.8mm-wide magnetic tape in a plastic cassette. (Digital audiotape.)
<b>Dead queue</b>	A queue containing stories that have been either deleted by users or purged automatically by the system. These stories are recycled automatically as new space is required.
<b>device</b>	Any computer peripheral or hardware component (such as printer, mouse, monitor, or hard disk) capable of receiving or sending data.
<b>dialog box</b>	A secondary window that gathers additional information from a user. It usually contains only a close (X) button in the top right corner, and can be removed from the screen by pressing the Escape (ESC) key. See also <i>window</i> .
<b>directory</b>	Like a file drawer in a file cabinet, a directory is a storage space. Directories, also known as <i>folders</i> , can contain other directories (known as sub-directories) or queues. Directories do not contain stories.
<b>Directory panel</b>	An area in the Avstar Workspace that displays the hierarchy of directories (folders) and queues in the Avstar database. Users can use the Directory panel to navigate through the system.

<b>drop-down menu</b>	A menu that is displayed from a menu bar—the bar of words, such as File or Edit, located at the top of a window. The menu can appear as a result of a mouse click on the menu bar or a keystroke combination of ALT plus the underlined letter of a word in the menu bar. For instance, ALT-F will open the File drop-down menu. See also <i>pop-up menu</i> .
<b>duration</b>	The length of a show or story. It is calculated by using the elapsed time in a broadcast when a story begins.
<b>easy lock</b>	A feature that allows a user to open a queue or story while preventing others from doing the same. It is similar to a key lock, but is created without a key. Therefore, others cannot be granted access. See also <i>lock, key lock</i> .
<b>Edit Decision List (EDL)</b>	A list of edits made during offline editing and used to direct the online editing of the master.
<b>edit lock</b>	A feature that prevents two people from working in a story simultaneously. The Avstar system automatically places a story in edit-lock mode when a user is working in a story, and a user can also manually edit lock a story.
<b>encode</b>	The process of converting analog video to a digital form.
<b>Ethernet</b>	A standard for connecting computers in a local area network (LAN). The actual technicalities are based on a Collision Sense Multiple Access protocol (CSMA).
<b>export</b>	<ol style="list-style-type: none"><li>1. To create an EDL from a sequence.</li><li>2. To conform a sequence.</li></ol>
<b>extract</b>	To remove a selected area from an edited sequence and close the resulting gap in the sequence.
<b>float</b>	To temporarily suspend a story. The story's time is removed from the show timing. Float time is also ignored by the teleprompter and

---

	machine control. Floating is used when you are not sure whether or where to put a story in a rundown.
<b>folder</b>	See <i>directory</i> .
<b>form</b>	A preformatted layout (template) containing the fields and the field positions (such as presenter and writer) required for a story. The form serves as a copy master when creating a new story.
<b>gigabyte (GB)</b>	Approximately one billion bytes (1,073,741,824 bytes) of information.
<b>grommet</b>	The indicator in a Story Text panel that links a script to production information. Also called: <i>anchor</i> , <i>grotch</i> , or <i>marker</i> .
<b>hard out</b>	A story in a newscast that has a fixed start time, usually at the end of a segment or show. It is manually entered into the system.
<b>headframe</b>	A single frame that can be used to help visually identify a clip, a sub-clip, or a sequence.
<b>high-resolution</b>	Digital video of a resolution suitable for broadcast.
<b>IN point</b>	Starting point of an edit. Also known as a <i>mark IN</i> .
<b>insert</b>	The process of including a subclip into a sequence.
<b>Instruction panel</b>	The area of the Story panel that contains production cues or machine control data. This area can be removed (hidden) from the display within the Story panel, so it may not appear on screen. See also <i>Story panel</i> , <i>Story Form panel</i> , <i>Story Text panel</i> .
<b>IP address</b>	Internet Protocol address is a 32-bit numeric identifier usually expressed as four groups of 8-bit decimal numbers (0 to 256) separated by dots, as in 192.168.0.1.

<b>ISA</b>	Industry Standard Architecture. A bus standard used in personal computers.
<b>key</b>	A special alphanumeric code that a user assigns to a queue or story to lock it. To open, or unlock, a queue or story, a user must have the key. See also <i>lock, easy lock, key lock</i> .
<b>key lock</b>	A feature that allows a user to lock a queue. To open the key-locked queue, all users (including the individual who put the key lock on the queue) must know the "key" if they want to open, move, duplicate, print, or delete the queue. See also <i>lock, easy lock</i> .
<b>kill</b>	To delete a story and place it in the Dead queue.
<b>lineup</b>	See <i>rundown</i> .
<b>load</b>	The process of opening a clip into the editor in preparation for viewing or editing.
<b>Local Area Network (LAN)</b>	This is a network of computers located in a common environment, such as in a building or building complex.
<b>lock</b>	To protect a queue or story from access by unauthorized users. A queue or story can be locked and unlocked with a key or by a user-name specific lock. See also <i>key, easy lock, key lock</i> .
<b>low-resolution</b>	Digital video of a resolution suitable for edits.
<b>marker</b>	<ol style="list-style-type: none"><li>1. The indicator in a Story Text panel that links a script to production information. Also called: <i>grommet, grotch, or anchor</i>.</li><li>2. A mark added to a selected frame to qualify a particular location within a sequence.</li></ol>
<b>menu</b>	See <i>drop-down menu, pop-up menu</i> .



---

<b>Messages of the Day window</b>	A window that displays one or more messages for Avstar system users when they log in to the system.
<b>mirroring</b>	A fault tolerance method that keeps identical copies of data on disk partitions located on different physical hard disks and servers.
<b>multimedia</b>	In computing, multimedia refers to the presentation of information on a computer using sound, graphics, animation, and text.
<b>network</b>	A group of computers and other devices connected together so they can communicate with each other.
<b>network address</b>	A network number that uniquely identifies a network cable segment. It is also referred to as the IPX external network number.
<b>NRCS</b>	An acronym for Newsroom Computer System.
<b>order lock</b>	A temporary lock that the Avstar system places on a queue when a user changes a sequence of stories in that queue. Order locking does not prevent other users from accessing the queue, but does prevent them from ordering the queues simultaneously.
<b>OUT point</b>	End point of an edit, or a mark on a clip indicating a transition point. Also known as a <i>mark OUT</i> .
<b>out time</b>	The total length of time for a show (shown in hours, minutes, and seconds) or the actual time by which the show must end (shown in 12-hour-clock time). See also <i>backtime</i> .
<b>panel</b>	A part of an Avstar Workspace. In Avstar, the three panels are the Directory panel, Queue panel, and Story panel. The Story panel is split further into the Story Form panel, Story Text panel, and Instruction panel (used for production cues).
<b>partition</b>	A method of assigning disk space, creating two or more virtual disks from a single physical disk. Similar to creating a directory.

<b>password</b>	A word users enter when logging in to the Avstar system. Passwords are alphanumeric and must be between 5- and 12-characters long.
<b>pathname</b>	The hierarchical name of the directory and queue in which a story is located. For instance, the pathname for the Yankees queue is WIRES . SPORTS . STORIES . YANKEES.
<b>PCI</b>	Peripheral Component Interconnect. A bus standard used in newer computers.
<b>player controls</b>	The electronic equivalent of a tape-deck controls.
<b>pop-up menu</b>	A menu that appears at the mouse pointer location when a user executes a right-click on a selected object. It contains commands contextually relevant to the selection. Also known as context menu. See also <i>drop-down menu</i> .
<b>presenter</b>	The person presenting a newscast on-air to a television audience. Also called a <i>presenter</i> .
<b>priority queue</b>	<ol style="list-style-type: none"><li>1. An area where Avstar places copies of wire stories (usually in WIRES . ADVISORY . PRIORITY).</li><li>2. IA queue designated to be read first by a server program for new stories.</li></ol>
<b>production cue</b>	A prompt to start a story element, such as a video playback. In Avstar, it is typically to the left of a scripted story, and provides information for other devices being controlled in the rundown. Production cues are usually prefaced by an asterisk (*). See also <i>marker</i> .
<b>purge</b>	To remove stories from queues (based on age) and place them in the DEAD queue. Purged stories are recycled as needed as new space is required. See also <i>purge interval</i> .

---

<b>purge interval</b>	A queue trait that indicates the time after which a story is considered “old.” Avstar will scan the entire database hourly and purge old stories from a queue.
<b>queue</b>	An area of the database that contains stories related to a general topic. Like a folder in a file drawer, queues are storage places within a system’s file structure that allow you to organize information in detailed categories. See also <i>directory</i> , <i>Directory panel</i> , <i>Queue panel</i> .
<b>queue form</b>	The area used to display the contents, size, and labels of a Queue panel.
<b>Queue panel</b>	An area in the Avstar Workspace that contains a list of the stories in a queue. Users can add, delete, and sequence stories in the Queue panel.
<b>queue property</b>	A trait that controls the characteristics of a queue in the Avstar database. Queue properties include the refresh trait, read-only purge interval, sorting, and so forth.
<b>read access</b>	Authority granted to users that allows them to read and duplicate the contents of a directory, queue, or story.
<b>read rate</b>	The number of words per minute at which a talent can read a news story. The system determines the total running time of a newscast based on the read rate of the assigned presenter.
<b>refresh</b>	A queue property or trait that automatically updates your screen’s display of the queue when changes are made to that queue by another user or by the system.
<b>relative-to-mark time</b>	Time is displayed as though the start of the clip is at the locator mark.
<b>relative-to-start time</b>	Time is displayed as though the start of the clip is at 00;00;00;00.

<b>remote service</b>	An archival system, bulletin board, or any information service that allows you to establish a connection to another service.
<b>results queue</b>	An area in Avstar in which the results from a Find All search are placed.
<b>roll</b>	To play a video. The digital equivalent of starting the tape deck.
<b>RS-232</b>	The Electronic Industries Association standard for short-range serial control.
<b>RS-422</b>	The Electronic Industries Association standard for medium-range serial control.
<b>rundown</b>	A lineup or timed-out list of stories indicating the order in which they will be aired during a news program. A rundown is viewed in the Queue panel of the Avstar Workspace. The rundown queue typically uses a form with BACK-TIME or CUME-TIME fields to display the timing of the newscast.
<b>SCO</b>	Santa Cruz Operation UNIX operating system.
<b>scratch pad</b>	A buffer in which text or notes are stored until the appropriate recovery procedure is performed. Deleted text and notes are stored in the scratch pad. It is separate from the Windows Clipboard and allows clippings to be accumulated.
<b>script</b>	A story that is read on the air. Typically, a script also contains production cues and references to the related media annotations.
<b>SCSI</b>	Small Computer System Interface is an industry standard with guidelines for connecting peripheral devices (such as hard drives and tape backup systems) and their controllers to a microprocessor. The SCSI, commonly pronounced "scuzzy," interface defines standards for hardware and software to communicate between a host computer and a

peripheral device. Computers and peripheral devices designed to meet SCSI Specifications are normally compatible.

<b>selection bar</b>	The box at the left edge of a Queue panel that, when clicked, selects a story and all of that story's details.
<b>server</b>	<ol style="list-style-type: none"><li>1. A special utility program the system uses to handle the distribution of stories internally. Some of these servers are known as: action, distribution, parallel, and so forth.</li><li>2. Computer hardware (or file servers) with the Avstar database and running the Avstar application software. These computers also run other operating systems, such as UNIX or Windows NT. For instance, the FTS server and the Avstar servers (also called server A, server B, or AVSTAR-A, AVSTAR-B, and so forth).</li></ol>
<b>session</b>	The way in which an Avstar Workspace is customized. Toolbars, workspace layout, and preferences can be customized and saved with a session.
<b>slave printer</b>	A printer attached to the workstation.
<b>sorted queue</b>	A queue in which stories are sorted according to criteria specified by the system administrator.
<b>source queue</b>	A queue from which stories are copied or moved.
<b>story</b>	Uniquely identified file containing text; stories are grouped in queues and are displayed in the Story panel of the Avstar Workspace.
<b>Story Form panel</b>	An area at the top of a Story panel that contains information about a story, such as its title, length, or status. This information is provided in fields that may correspond to data displayed in the Queue panel. This area can be removed (hidden) from the display within the Story panel, so it may not appear on screen.

<b>Story panel</b>	An area in the Avstar Workspace that displays the story form, text, and production cues of a story.
<b>Story Text panel</b>	An area in the Story panel that contains the text or script of a story. It is the only area that is always displayed as part of the Story panel. See also <i>Story panel</i> , <i>Story Form panel</i> , <i>Instruction panel</i> .
<b>subclip</b>	An edited part of a clip. In a sequence, a subclip is bound by <i>mark IN</i> and <i>mark OUT</i> points.
<b>superuser</b>	A user account that is given access to restricted functions in the Avstar system. Only a system administrator can assign superuser status.
<b>system administrator</b>	A person responsible for maintaining the Avstar system and keeping all functions operating properly.
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol is a platform-independent protocol for intercomputer communication.
<b>time bar</b>	A graphical representation of the duration of a clip, including an indication of the current position and the IN and OUT marks.
<b>user</b>	Individual who has a valid user account in the Avstar system.
<b>user ID</b>	A special alphanumeric code that identifies a user account in the Avstar system. A user ID can be up to 20-characters long.
<b>user manager</b>	User ID given the authority to add, modify, delete, and search for information about user accounts. User manager status can be assigned by a system administrator only.
<b>user name</b>	A word established to identify the individual user. Enter your user name and your password to log in. User names are alphanumeric and are up to 20-characters long.

<b>video clip</b>	See <i>clip</i> .
<b>window</b>	An area in which the main interaction takes place. It is typically rectangular, with a title bar (that appears blue when active and gray when inactive), and three buttons in the top right corner (used to resize the window or close it). Unlike a dialog box, a standard window cannot be closed by pressing the Escape key.
<b>Windows</b>	Graphical shell operating environment that runs on top of DOS. It contains many accessories and features that access DOS functions such as file, program, and printer management. Windows is referred to as a GUI (Graphical User Interface).
<b>Windows NT</b>	Microsoft Windows New Technology Operating System that implements protected process multitasking, security, and other features of traditional operating systems, while maintaining a high level of compatibility with other Windows operating systems.
<b>wire bulletin</b>	See <i>bulletin</i> .
<b>workspace</b>	The area within the Avstar main window consisting of the Directory panel, Queue panel, and Story panel. The Avstar Workspace is where users can view, add, edit, and delete information.
<b>write access</b>	The ability to add new stories, edit existing stories in a particular queue, add a queue, or add a directory.





# Index

## Symbols

- : console prompt . . . . . 3-3
- # console prompt . . . . . 3-3
- + and - characters . . . . . 14-29
- @ (at) character . . . . . 2-4
  - in repeating macros . . . . . 7-21
- | (pipe) . . . . . 22-12

## Numerics

- 7-bit character set . . . . . E-2
- 8-bit characters in mail . . . . . 14-82

## A

- absolute date, defined . . . . . G-9
- abstract printing . . . . . 5-37
  - abstract lines database trait . . . . . G-29
  - abstract style database trait . . . . . G-30
- accent keyword . . . . . 13-13
- access, restricting read/write . . . . . 6-27
- account created option . . . . . 18-8
- account queue
  - form . . . . . 8-27
  - using the CONNECT command . . . . . 8-27
- action servers . . . . . 14-21 to 14-26
  - adding . . . . . 14-22 to 14-26
  - assigning distribution codes . . . . . 14-34
  - defined . . . . . 14-2
  - flow of events . . . . . 14-8
  - timed interval tasks . . . . . 14-20
  - validation . . . . . 14-26

## adding

- CCUs to the bootptab file . . . . . 11-32
- comments to a file . . . . . 11-10
- comments to macros . . . . . 7-6
- configuration lines . . . . . 11-15
- distribution lines . . . . . 13-35
- DOS PC . . . . . 11-39 to 11-41
- members to a group . . . . . 6-15
- network links
  - Rx and Tx link . . . . . 14-94 to 14-100
- network services . . . . . 17-6
- services . . . . . 17-5
- user accounts . . . . . 4-22 to 4-29
- adding devices . . . . . 11-29, 11-48
  - action servers . . . . . 14-22 to 14-26
  - CCU . . . . . 11-30
  - distribution server . . . . . 14-38 to 14-42
  - DOS or Windows PC . . . . . 11-39
  - keyword server . . . . . 14-51 to 14-60
  - network Rx and Tx links . . . . . 14-93 to 14-100
  - parallel wire server . . . . . 14-43 to 14-51
  - PCUs . . . . . 11-33
  - print servers . . . . . 14-79 to 14-80
  - printers . . . . . 11-41 to 11-47
  - Rx links . . . . . 14-93
  - seek servers . . . . . 14-61 to 14-63
  - servers . . . . . 14-22
  - to the configuration file . . . . . 11-15
  - TX links . . . . . 14-93
  - wire service . . . . . 11-47 to 11-48, 13-2 to 13-11
  - workstations . . . . . 11-35
- AFF-READY-N form field . . . . . 8-13
- air command, default margin . . . . . 11-21
- AIR-DATE form field . . . . . 8-14
- airmargin system profile parameter . . . . . 11-21
- airmargin, printer profile . . . . . 12-14
- airshift system profile parameter . . . . . 11-21
- alias
  - adding nonprinting characters . . . . . 12-19
  - for mail . . . . . 6-28 to 6-31
  - nonprinting characters . . . . . 13-26

- ALWAYS option . . . . . 13-35  
 anchored element... *See production cue set story entity*  
 AND search rule . . . . . 13-45  
 ANPA character set . . . . . 13-30  
 answer keyword . . . . . 13-13  
 answerback, printer profile . . . . . 12-8  
 APP1-N to APP5-N form field . . . . . 8-14  
 arabic  
     character conversion. . . . . E-15  
     wire profile. . . . . E-15  
 ARCHIVE.PACKAGES queue. . . . . 14-93  
 ARCHIVE.SCRIPTS queue . . . . . 14-93  
 ASCII tables . . . . . E-2  
 assigning  
     commands to a function key . . . . . 2-11  
     distribution codes. . . . . 14-30, 14-32  
     form field validation. . . . . 14-26  
     keyboards. . . . . 7-16  
     mailboxes to a queue . . . . . 14-17  
     mailboxes to a queue at console . . . . . G-31  
     read and write permissions . . . . . 6-22 to 6-27  
     VT macros to standard macro keys . . . . . 7-22  
 at command . . . . . 14-19  
 at job list command . . . . . A-42  
 AUDIO-TIME form field . . . . . 8-15  
 auto\_upgrade system profile parameter . . . . . 11-21  
 autocue printer profile. . . . . 12-8  
 autofield, printer profile . . . . . 12-8  
 autoindent, printer profile . . . . . 12-8  
 autolength, printer profile . . . . . 12-8  
 automatic time-out . . . . . 6-19  
 autopindent, printer profile . . . . . 12-8  
 autoshift, printer profile . . . . . 12-9  
 autospace, printer profile. . . . . 12-8  
 Avstar console  
     at character (@) . . . . . 2-4  
     control commands . . . . . 2-3  
     PRINTER display . . . . . 2-22  
     zoom in on one server . . . . . 2-7  
 Avstar NRCS  
     described . . . . . 1-4  
     Ethernet address list . . . . . 11-39  
     programs used by . . . . . A-2  
     sample layout. . . . . 1-7  
     shutting down the system . . . . . 3-9 to 3-12  
     starting the system. . . . . 3-5 to 3-9  
     UNIX commands available . . . . . A-4  
 Avstar Systems, products . . . . . 1-3
- ## B
- backspace key . . . . . 2-4  
 BACK-TIME form field . . . . . 8-15  
 backup  
     appending to tape . . . . . 20-8  
     continuation tape. . . . . 20-9  
     data to tape . . . . . 20-3 to 20-9, 20-20 to 20-22  
     database. . . . . 20-5  
     dbdump command . . . . . 20-4  
     individual queues . . . . . 20-7  
     listing tape backup dates . . . . . 20-11, 20-13  
     listing tape contents . . . . . 20-11  
     maximum stories found. . . . . 20-17  
     new tape . . . . . 20-7  
     procedures . . . . . 20-2  
     queues . . . . . 20-3, 20-7, 20-20  
     retrieve stories from tape. . . . . A-12  
     searching  
         a tape . . . . . 20-14  
         by word and date . . . . . 20-16  
         for stories by words . . . . . 20-15  
         for stories, word and day . . . . . 20-17  
         for stories, word and month. . . . . 20-17  
     site files to tape . . . . . 20-25  
     sitedump command. . . . . 20-25, A-35  
     skip flag . . . . . 20-4  
     software . . . . . 20-22  
     system . . . . . A-35  
     system files . . . . . 10-2

- backup (continued)
    - tape . . . . . A-8
    - verifying . . . . . 20-5, 20-8
    - workstation, automatic . . . . . 4-14
  - bad configuration message . . . . . 11-16
  - bad story, fix link counts . . . . . A-16
  - banner, printer profile . . . . . 12-13
  - baud rate, printers . . . . . 11-43
  - bell keyword. . . . . 13-13
  - bitmaps for CG templates. . . . . 9-6
  - bits keyword. . . . . 13-13
  - blank command for VT. . . . . 7-20
  - blanks, printer profile . . . . . 12-13
  - blocks
    - database organization . . . . . 19-3
    - locked . . . . . 22-11
  - body story entity . . . . . 15-5
  - bootptab file
    - adding a CCU . . . . . 11-32
    - adding a PC on SCO UNIX systems . . . . . 11-33
  - bottom console control command . . . . . 2-21
  - broadcast command . . . . . A-4
    - defined . . . . . 3-10
    - message. . . . . A-4
  - Broadcast Control System (BCS), described . . . . . 1-6
  - broadcast message . . . . . A-4
  - browsing the news database. . . . . 16-1
  - bscan command . . . . . 14-9, A-42
  - building dialogs . . . . . 17-2
  - BULLETIN constraint level . . . . . 13-34
  - busy stories. . . . . 22-6
    - introduction . . . . . 5-52
- C**
- CA-CAPTURED form field . . . . . 8-15
  - CA-DIRECTION form field . . . . . 8-15
  - CA-ELAPSED form field . . . . . 8-16
  - CA-IDENT form field. . . . . 8-16
  - cancelling print jobs . . . . . 12-48
  - CA-ORIGIN form field. . . . . 8-16
  - capture dialog command . . . . . A-48
  - capture tool . . . . . 9-6 to 9-10
    - installation . . . . . 9-8
  - CA-RECEIVED form field . . . . . 8-16
  - CA-REMOTE form field. . . . . 8-16
  - CA-SENT form field. . . . . 8-16
  - cat command . . . . . 10-2, 11-18, 11-45
    - wire profile . . . . . 13-12
  - categories, hidden . . . . . 13-35
  - category and keyword check
    - program messages . . . . . C-10
  - category keyword . . . . . 13-15
  - cccolor . . . . . F-3
  - CCU
    - adding . . . . . 11-30
    - adding to the bootptab file . . . . . 11-32
    - ccureset command . . . . . A-32
    - configuration . . . . . 11-31
    - connecting devices . . . . . D-4
    - device numbering conventions. . . . . 11-34
    - environment . . . . . D-5
    - Ethernet . . . . . D-3
    - introduction . . . . . D-2
    - messages dictionary . . . . . C-15
    - moving a printer. . . . . 11-42
    - network . . . . . D-3
    - parameter, list c command . . . . . 11-6
    - specifications . . . . . D-5
  - CCU II
    - back panel. . . . . D-8
    - connecting devices . . . . . D-8
    - error light . . . . . D-8
    - front panel . . . . . D-7
    - I/O ports . . . . . D-8
    - power light . . . . . D-7
    - reset switch. . . . . D-7
  - CCU III
    - back panel. . . . . D-14
    - common backplane design . . . . . D-15
    - connecting to computer . . . . . D-15

- CCU III (continued)
  - error codes . . . . . D-12
  - front panel . . . . . D-11
  - I/O ports . . . . . D-14
  - LED display . . . . . D-11
  - plug-in cards . . . . . D-15
  - reset switch. . . . . D-10
- CCU IV
  - back panel . . . . . D-19
  - common backplane design. . . . . D-20
  - connecting to computer . . . . . D-20
  - connecting to server . . . . . D-20
  - front panel . . . . . D-17
  - I/O ports . . . . . D-19
  - LED display . . . . . D-18
  - plug-in cards . . . . . D-20
  - reset button . . . . . D-16
- ccucmds . . . . . C-2
- ccuputkey console command. . . . . A-4
- ccureset console command . . . . . A-32
- CG template editor . . . . . 9-11 to 9-22
- CG templates
  - backgrounds. . . . . 9-6
  - creating. . . . . 9-15
  - explained . . . . . 9-4
- CG title entry
  - access . . . . . 9-23
  - capture tool . . . . . 9-6 to 9-10
  - dialog box. . . . . 9-3
  - explained . . . . . 9-2
  - installing capture tool. . . . . 9-8
  - read only option. . . . . 9-19
  - security. . . . . 9-23
  - setup. . . . . 9-4 to 9-10
- CG-ID form field . . . . . 8-17
- CG-TEMPLATE form field . . . . . 8-17
- CG-TEXT form field. . . . . 8-17
- changing
  - command assignment of a function key. 2-11
  - database traits . . . . . 5-21
  - dictionary values . . . . . C-4
  - directories and queues, security. . . . . 5-21
  - names for directories and queues . . . . . 5-10
  - the configuration file . . . . . 11-15
  - user passwords . . . . . 18-3
- CHANNEL form field. . . . . 8-17
- character conversion . . . . . 14-84
- character conversion table, mail . . . . . 14-83
- character mapping
  - dbrestore . . . . . A-13
  - devices . . . . . A-51
  - service . . . . . A-51
  - VT . . . . . C-46
- character set
  - ANPA . . . . . 13-30
  - ISO . . . . . 13-30
- characters
  - by alias or by decimal value. . . . . 12-20
  - used by the system. . . . . 12-20
- checking for a locked block. . . . . 22-11
- checking the database for errors . . . . . 19-8
- choosing device numbers . . . . . 14-94
- cleaning the database
  - monthly . . . . . 19-9 to 19-13
  - offline. . . . . 19-10 to 19-13
- clockmax system profile parameter. . . . . 11-21
- combining security levels . . . . . 6-17
- command key . . . . . 2-3
- command line, distribution codes. . . . . 14-34
- command set. . . . . 14-10
- commands
  - assigning to function keys . . . . . 2-11
  - ccuputkey . . . . . A-4
  - changing the title of . . . . . C-4
  - dialog. . . . . A-48
  - hiding. . . . . C-4
  - iNews personnel only . . . . . A-3
- commands dictionary . . . . . C-19
- comments, adding to macros . . . . . 7-6
- communication, testing . . . . . A-32

- computer (server)
  - defined . . . . . 11-16
  - name . . . . . 11-25
  - selecting . . . . . 2-5
  - selecting using zoom. . . . . 2-6
- computer commands . . . . . 2-4, 2-21
- COMPUTER parameter, list c command . . . 11-6
- configuration file
  - adding a configuration line. . . . . 11-15
  - adding comments . . . . . 11-10
  - CCU line . . . . . 11-31
  - changing (caution) . . . . . 11-8
  - checking for errors . . . . . 13-9, A-5
  - comments . . . . . 11-13
  - debugging. . . . . 11-16
  - defined . . . . . 11-1
  - device numbers and CCU ports . . . . . 11-31
  - device section, or body . . . . . 11-13
  - editing in the database . . . . . 11-48
  - host section . . . . . 11-13
  - incorporating changes. . . . . 11-10, A-5
  - listing contents . . . . . 11-10
  - location . . . . . 11-10
  - making changes . . . . . 11-15
  - modifying a configuration line . . . . . 11-15
  - network Tx link. . . . . 14-98
  - print servers . . . . . 14-79
  - printers . . . . . 11-42, 11-44, 13-4
  - punctuation/programming function . . 11-10
  - testing after changes . . . . . 11-9
  - wires . . . . . 13-2, 13-3
  - workstations . . . . . 11-36
- configuration keywords . . . . . 2-19
- configure command . . . . . 11-16, 13-9, A-5
- configure console command. . . . . 11-9
- configuring devices
  - licensing categories . . . . . 11-2
  - printers . . . . . 11-41 to 11-47
- connect
  - changing escape character. . . . . A-50
  - dictionary . . . . . C-33
  - idle time-out . . . . . 11-27
  - installing a service . . . . . 17-5
  - mapping characters . . . . . A-51
- connect command . . . . . A-6
  - waiting for . . . . . 17-3
- connecting a service . . . . . 17-5
- connecting devices
  - multiple servers . . . . . 14-88
  - network resources . . . . . 17-6
  - printers . . . . . 11-41 to 11-47
- connectivity, testing with ping. . . . . 22-15
- console
  - COMMAND prompt . . . . . 2-4
  - dialing in to. . . . . 2-14
  - executing workstation commands . . . . A-4
  - exiting. . . . . 2-13
  - freezing. . . . . 2-12
  - keyboard. . . . . 2-4
  - logging in . . . . . 3-2
  - managing groups . . . . . G-34
  - overview. . . . . 2-1 to 2-2
  - passwords. . . . . 3-3 to 3-5
  - pausing screen display . . . . . 2-8
  - printing history. . . . . 2-22
  - remote. . . . . 2-14 to 2-16
  - starting . . . . . 2-14
  - time display . . . . . 11-29
- console command
  - broadcast . . . . . 3-10, A-4
  - ccureset. . . . . A-32
  - configure. . . . . A-5
  - connect . . . . . A-6, A-33
  - dbclean . . . . . A-7
  - dbclean -x . . . . . 3-8
  - dbclose . . . . . A-8
  - dbdev . . . . . A-8
  - dbdump . . . . . A-8
  - dbfree . . . . . A-10
  - dblines . . . . . A-10
  - dboriginal. . . . . A-11
  - dbpurge . . . . . A-11

## console command (continued)

- dbrestore . . . . . 20-10, A-12
- dbserver . . . . . A-14
- dbsort . . . . . A-14
- dbvisit . . . . . A-16
- described . . . . . 2-3
- dial . . . . . A-17
- diskclear . . . . . A-18
- diskcopy . . . . . A-18
- doc . . . . . A-19
- edit . . . . . A-19
- entering . . . . . 2-3
- extract . . . . . A-19
- force . . . . . 18-5, 18-7, A-20, G-5
- grpcheck . . . . . A-20
- gtraits . . . . . A-20
- help . . . . . A-21
- hogs . . . . . A-21
- list a\* . . . . . A-22
- list c . . . . . A-22
- list d . . . . . A-23
- list g . . . . . A-23
- list s . . . . . A-24
- list u . . . . . A-25, G-10
- logout . . . . . A-26
- ls . . . . . A-27
- makeccutab . . . . . A-28
- maketab . . . . . A-29
- msgclean . . . . . A-30
- number . . . . . A-30
- offline . . . . . 3-10, A-31
- online . . . . . A-31
- otod . . . . . A-31
- passwd root . . . . . 3-4
- print . . . . . A-32
- ps . . . . . 11-20
- reconnect . . . . . A-33
- reference table . . . . . 2-21
- rename . . . . . A-33
- restart . . . . . A-34
- rmdir . . . . . A-34
- searchtape . . . . . A-35
- send . . . . . A-35
- shutdown . . . . . A-35
- sitedump . . . . . A-35
- siterestore . . . . . A-36
- softdump . . . . . A-36
- softrestore . . . . . A-36
- startup . . . . . A-36
- status . . . . . 11-20
- stop . . . . . A-38
- su . . . . . A-38
- utraits . . . . . A-39
- version . . . . . A-40
- wholockedit . . . . . A-40
- wiredump . . . . . A-41
- console configuration file . . . . . 2-16 to 2-20
- console configuration keywords . . . . . 2-19
- console control commands... *See console command*
- console conventions, in this manual . . . . . xxx
- console history . . . . . 2-7 to 2-13
  - commands . . . . . 2-9
- console operations . . . . . 2-12
- console superuser . . . . . 3-2 to 3-3
  - exiting . . . . . 3-3
  - logging in . . . . . 3-3
- console.cfg system file . . . . . 2-16 to 2-20, B-10
- control codes . . . . . 12-4
  - forms . . . . . 12-6
- converting text . . . . . 13-28
- convertvideo . . . . . 12-29
- copying stories . . . . . 19-4
- cp command . . . . . 10-2
  - modifying a profile . . . . . 13-6
- CREATE-BY form field . . . . . 8-17
- CREATE-DATE form field . . . . . 8-18
- creating
  - areas in the news database . . . . . 4-23
  - database manager account . . . . . 4-36
  - folders/directories . . . . . 4-23 to 4-24, 5-4 to 5-6
  - forms . . . . . 8-2 to 8-10
  - HTML story forms . . . . . 15-5 to 15-23

- creating (continued)
    - keyboard stories . . . . . 7-3
    - multiple groups. . . . . 6-14
    - printer profiles . . . . . 11-44
    - queues, creating . . . 4-24 to 4-25, 5-6 to 5-7
    - stories, queue restrictions . . . . . 6-21
    - user manager accounts . . . . . 4-35
  - CUME-TIME form field . . . . . 8-18
  - current print request. . . . . 12-47
  - customizing
    - dictionaries . . . . . C-4
      - introduction. . . . . C-2
    - keyboards . . . . . 7-3
    - print effects . . . . . 12-4 to 12-5
- D**
- D flag, duplicated stories . . . . . 5-19
  - DAT tapes. . . . . 20-2
  - database
    - backing up the entire. . . . . 20-5
    - blocks . . . . . 19-3
    - checking for errors . . . . . 19-8
    - cleaning monthly . . . . . 19-9 to 19-13
    - cleaning offline . . . . . 19-10 to 19-13
    - cleanup . . . . . 19-9, 19-10
    - error check . . . . . A-16
    - free up space . . . . . G-28
    - purge interval . . . . . 19-2
    - purging cycle. . . . . 19-2
    - retrieve stories from tape . . . . . A-12
    - size . . . . . A-10
    - understanding storage . . . . . 19-3
  - database cycle. . . . . 19-2
  - database maintenance . . . . . 19-8
    - dbvisit . . . . . A-16
    - regular repair . . . . . A-16
  - database manager account, creating. . . . . 4-36
  - database security . . . . . 18-1
  - database space
    - increasing . . . . . 19-7
    - tracking. . . . . 19-5
  - database storage
    - available memory . . . . . 11-20
    - checking high water mark . . . . . 11-20
    - create space by purging . . . . . G-28
    - maintaining . . . . . 19-8 to 19-13
    - units . . . . . 19-3
  - database traits
    - abstract lines. . . . . G-29
    - abstract style. . . . . G-30
    - apply changes to all directories/queues 5-24
    - display lines . . . . . G-32
    - display lines for rundown queue . . . . . G-32
    - forms. . . . . 8-2
    - general . . . . . 6-21
    - lockuser . . . . . 5-48
    - notification group . . . . . 6-25
    - orderuser . . . . . 5-48
    - parent database traits . . . . . 5-21
    - parent dbtraits at console . . . . . G-17
    - purge interval. . . . . 5-41, G-27
    - read group . . . . . 6-23
    - reducing purge interval . . . . . 5-41
    - refresh. . . . . G-34
    - skip flag . . . . . 20-4
    - sortfield. . . . . G-25
    - summary. . . . . 5-25 to 5-47
    - using with user traits . . . . . 5-3
    - viewing. . . . . 5-13 to 5-21
    - write group. . . . . 6-24
  - date1 and date2 characters. . . . . G-9
  - dbclean console command . . . . . A-7
  - dbclose console command . . . . . A-8
  - dbdev console command . . . . . A-8
  - dbdump c command . . . . . 20-5
  - dbdump command . . . . . 20-4, 20-7
    - uses of. . . . . 20-3 to 20-9
  - dbfree command . . . . . 20-18, A-10
    - space usage. . . . . 19-6
  - dblines command. . . . . 19-8, A-10

- dbmanager, CG title entry . . . . . 9-23
- dbmanager account, creating . . . . . 4-36
- dboriginal console command . . . . . A-11
- dbpurge command . . . . . 19-2, A-11
  - space usage . . . . . 19-6
- dbrestore, character mapping . . . . . A-13
- dbrestore command . . . . . 20-8, 20-10, A-12
- dbrestore tdv command . . . . . 20-5
- dbserver console command
  - and purge limit . . . . . 11-26
  - syntax . . . . . A-14
- dbserver program messages . . . . . C-9
- dbsort console command . . . . . A-14
- dbtraits . . . . . 5-10
- dbvisit . . . . . 5-10
- dbvisit command . . . . . A-16
  - checking the database . . . . . 19-11
- Dead queue . . . . . 5-19, 19-2
  - introduction . . . . . 19-2
  - reclaim space . . . . . A-14
  - reclaiming . . . . . 19-6
  - recovering stories from . . . . . 5-20
  - retrieving a story . . . . . 19-2
  - unbusyng . . . . . 22-7
- debugging the configuration file . . . . . 11-16
- decimal values
  - adding to a printer profile . . . . . 12-20
  - nonprinting characters . . . . . 13-27
- DECnet, using with Rx/Tx links . . . . . 14-94
- default dictionary, changing values . . . . . C-4
- defining
  - a queue, scan line . . . . . 14-9
  - fonts . . . . . 12-4
  - groups . . . . . 6-5
  - messages/commands using dictionaries . . . . . C-2
- delay dialog command . . . . . A-49
- delete\_notify . . . . . F-19
- deleted stories, processing . . . . . 14-12
- deleting
  - function key command assignments . . . . . 2-11
  - groups . . . . . 6-13
  - runaway print jobs . . . . . 22-10
  - stories . . . . . 14-11
- designating the master computer . . . . . 11-24
- destination queue . . . . . 13-33
  - user traits, destination queue . . . . . 4-7
- destinationorder . . . . . F-5
- DEV parameter, list c command . . . . . 11-5
- device # parameter . . . . . 11-36
- device name parameter . . . . . 11-37
- device number for the printer . . . . . 11-41
- device numbering . . . . . 14-94
- device parameter
  - order in CCU configuration line . . . . . 11-31
- DEVICE\_TYPE parameter, list c command . . . . . 11-6
- devices
  - adding . . . . . 11-29 to 11-48
  - adding or removing . . . . . 11-15
  - character mapping . . . . . A-51
  - connecting to CCU . . . . . D-4
  - definition . . . . . 11-1
  - numbering conventions . . . . . 11-34
  - servers . . . . . 11-3, 14-1
  - types . . . . . 11-3
  - viewing information about . . . . . 11-4 to 11-7
- DEVNAME parameter, list c command . . . . . 11-7
- diag dialog command . . . . . A-49
- dial command . . . . . A-17
- dialback option . . . . . 18-2
- dialing in to the console . . . . . 2-14
- dialog box
  - CG title entry . . . . . 9-3
  - change user's password . . . . . 4-9
  - local printing . . . . . 12-32 to 12-37
  - manage font presets . . . . . 9-21
  - modify user account . . . . . 4-4 to 4-9
  - simplified user setting . . . . . 4-20 to 4-22
  - user preferences . . . . . 4-11 to 4-19



- dialog commands
  - capture . . . . . A-48
  - delay . . . . . A-49
  - diag . . . . . A-49
  - echo . . . . . A-49
  - escape . . . . . A-50
  - expect . . . . . A-50
  - heol . . . . . A-51
  - introduction . . . . . A-48
  - list . . . . . A-48
  - map . . . . . A-51
  - mapin . . . . . A-52
  - mapout . . . . . A-52
  - message . . . . . A-53
  - pass . . . . . A-53
  - pause . . . . . A-54
  - stop . . . . . A-54
  - timer . . . . . A-54
  - type . . . . . A-55
  - wait . . . . . A-55
- dialog for service . . . . . 17-8
- dialogs . . . . . 17-2 to 17-4
  - building . . . . . 17-2
  - capture command . . . . . 17-2
  - creating a queue . . . . . 17-3
- dictionaries
  - building . . . . . C-6
  - case-shifting . . . . . C-44
  - CCU messages dictionary . . . . . C-15
  - commands dictionary . . . . . C-19
  - connect dictionary . . . . . C-33
  - customizing . . . . . C-4
  - defining messages and commands . . . . . C-2
  - dial dictionary . . . . . C-39
  - editing . . . . . C-5
  - incorporating changes . . . . . C-6
  - keyboard macros . . . . . C-40
  - printer messages dictionary . . . . . C-43
  - queues dictionary . . . . . C-27
  - restoring standard translations . . . . . C-8
  - space limitations . . . . . C-3
  - standard name . . . . . C-3, E-8
  - table space exceeded . . . . . C-8
  - telex dictionary . . . . . C-37
  - translations . . . . . C-3, E-8
  - utility messages dictionary . . . . . C-9
  - video attributes dictionary . . . . . C-26
  - vtmap . . . . . C-46
  - words dictionary . . . . . C-29
- dictionary defaults, restoring . . . . . C-8
- dictionary files
  - file locations . . . . . C-2
  - filenames . . . . . C-2
- dictionary values
  - changing . . . . . C-4
  - space limitation . . . . . C-3
- directories
  - choosing a purge interval . . . . . 5-41
  - creating . . . . . 4-23 to 4-24, 5-4 to 5-6
  - hiding . . . . . 6-27
  - removing . . . . . 5-10
  - renaming . . . . . 5-10
- directory trait, update queue . . . . . 14-90
- disabling mail to all users . . . . . 14-81
- disabling priority notification . . . . . 13-34
- disaster recovery . . . . . 20-20 to 20-22
- disk drive
  - invalid block . . . . . 19-3
  - valid block . . . . . 19-3
- disk space, distribution and purging . . . . . 19-2
- diskclear command . . . . . A-18
- diskcopy command . . . . . A-18
- disks in use, list . . . . . A-8
- display line
  - rundown queue . . . . . G-32
  - setting . . . . . G-32
- display function key command assignment . . . . . 2-12
- distributing wire stories . . . . . 13-31
- distribution codes
  - action server . . . . . 14-34
  - ALWAYS option . . . . . 13-35
  - assigning . . . . . 14-30, 14-32

- distribution codes (continued)
- assigning with DUPLICATE or MOVE . 14-34
  - TRANSMIT option . . . . . 13-34
  - Tx link . . . . . 14-34
  - validation . . . . . 14-35
  - wire distribution. . . . . 14-35
- distribution job list command . . . . . A-43
- distribution keyword . . . . . 13-15
- distribution lines . . . . . 13-31
- adding . . . . . 13-35
  - maximum . . . . . 13-38
- distribution name . . . . . 13-32
- distribution queue . . . . . 14-39
- distribution servers . . . . . 14-30 to 14-41
- adding . . . . . 14-38 to 14-42
  - defined . . . . . 14-2
  - distribution codes. . . . . 14-30
  - distribution queue . . . . . 14-39
  - distribution story . . . . . 14-39
  - illustrated. . . . . 14-32
  - mailbox. . . . . 14-16
  - queues and traits . . . . . 14-42, 14-52
- distribution story . . . . . 14-39
- entering . . . . . 14-39
  - matching and case . . . . . 14-36
  - matching and order . . . . . 14-36
  - notes . . . . . 13-35
  - wildcards and destination queue . . . . . 14-33
- DM-NAME form field . . . . . 8-18
- doc command
- to transfer material. . . . . 11-48
- doc console command A-19
- done, printer profile. . . . . 12-9
- DOS PC, adding. . . . . 11-39 to 11-41
- down keyword console command. . . . . 2-21
- down number console command . . . . . 2-21
- dup (duplicate) command . . . . . 14-10
- assigning distribution codes. . . . . 14-34
- dup job list command . . . . . A-43
- duplicated a story, who last. . . . . 5-17
- E**
- echo dialog command . . . . . A-49
- ed (UNIX line editor)
- defined. . . . . 10-4
  - ed command . . . . . 10-5
  - editing commands . . . . . 10-9
  - launching. . . . . 10-5
  - quitting . . . . . 10-14
  - saving changes. . . . . 10-14
  - searching a file. . . . . 10-7
  - searching for text . . . . . 10-7
- edit and order locks. . . . . 3-8
- edit command . . . . . A-19
- edit lock
- introduction . . . . . 5-52
  - unbusyng . . . . . 22-6
- Edit Title Entry Template window . . . . . 9-11
- toolbars . . . . . 9-12
- edit, using to modify a configuration line. . . 11-15
- editing a wire profile. . . . . 13-6
- editing the /site/config file. . . . . 11-48
- editing the dictionary file . . . . . C-5
- 8-bit characters in mail . . . . . 14-82
- eightmail . . . . . 14-82
- ejectcode, printer profile . . . . . 12-9
- ejectcount, printer profile . . . . . 12-9
- enabling users to receive mail . . . . . 4-29
- end keyword . . . . . 13-16
- ENDORSE form field . . . . . 8-18
- entering console commands . . . . . 2-3
- environment variables. . . . . F-1
- cccolor . . . . . F-3
  - delete\_notify . . . . . F-19
  - destinationorder . . . . . F-5
  - eightmail . . . . . 14-82
  - maillookup. . . . . F-7
  - msgmailalert . . . . . F-8
  - picolor . . . . . F-10
  - scan codes . . . . . F-13
  - showtimingbar. . . . . F-12

## environment variables (continued)

synctoserver .....	F-16
vtcompatibility .....	F-18
eol keyword .....	13-16
eop keyword .....	13-17
eos keyword .....	13-17
Errno .....	11-55
error checking in keyboard stories .....	7-13
error checking, story .....	A-10
error messages, group checker .....	6-9 to 6-12
errors in the database .....	19-8
escape dialog command .....	A-50
/etc/bootptab system file .....	11-32, B-2
DEC/MIPS specific .....	B-4
SCO-UNIX specific .....	B-2
SGI-Irix specific .....	B-3
/etc/hosts system file .....	11-17, B-4
/etc/networks system file .....	B-5
Ethernet	
address list, DOS clients .....	11-39
introduction .....	D-3
T-connector .....	D-3
evaluation order .....	13-45
EVENT-DURATION form field .....	8-19
EVENT-EFFECT form field .....	8-19
EVENT-STATUS form field .....	8-19
EVENT-STYLE form field .....	8-19
every command .....	14-19
every job list command .....	A-43
excludedvideo system profile parameter ..	11-22
executing commands remotely .....	2-15
exiting console superuser .....	3-3
exiting the console .....	2-13
exlines, printer profile .....	12-9
expanded text mode, printer profile .....	12-10
expect dialog command .....	A-50
extended macro keys	
assigning VT macros .....	7-24
extended macro keys for VT .....	7-23
extract console command .....	A-19

**F**

F13 key on the VT220-style terminal .....	7-24
fast text search (FTS) .....	14-63 to 14-78
fgrep command .....	22-12
field IDs story entity .....	15-5
file numbering for printer profiles .....	12-2
first level directory, restoring from tape ..	20-10
flags keyword .....	13-18
FLASH constraint level .....	13-34
flash keyword .....	13-19
folders, creating .....	4-23 to 4-24, 5-4 to 5-6
font and form space .....	12-7
Font PreSets, using .....	9-20
font, printer profile .....	12-10
fonts	
combining print effects .....	12-4
defining .....	12-4
difference between forms and fonts 12-5	
selecting .....	12-26
selecting in style story .....	12-26
space limitations .....	12-7
video mode .....	12-26
footers	
grouping in queues .....	12-15
printing .....	12-15
user-selected .....	12-16
force console command .. 18-5, 18-7, A-20, G-5	
forcing password changes .....	18-6
form field validation .....	14-26
Form Fields	
AFF-READY-N .....	8-13
AIR-DATE .....	8-14
APPN-X .....	8-14
AUDIO-TIME .....	8-15
BACK-TIME .....	8-15
CA-CAPTURED .....	8-15
CA-DIRECTION .....	8-15
CA-ELAPSED .....	8-16
CA-IDENT .....	8-16
CA-ORIGIN .....	8-16

## Form Fields (continued)

CA-RECEIVED . . . . .	8-16	WRITER . . . . .	8-24
CA-REMOTE . . . . .	8-16	form fields... <i>See Form Fields</i>	
CA-SENT . . . . .	8-16	form keyword . . . . .	13-20
CG-ID . . . . .	8-17	form, printer profile . . . . .	12-10
CG-TEMPLATE . . . . .	8-17	format strings, optional . . . . .	15-6, 15-14
CG-TEXT . . . . .	8-17	forms	
CHANNEL . . . . .	8-17	account queue . . . . .	8-27
CREATE-BY . . . . .	8-17	assigning to a queue . . . . .	8-11
CREATE-DATE . . . . .	8-18	combining . . . . .	12-6
CUME-TIME . . . . .	8-18	consistent naming . . . . .	12-6
DM-NAME . . . . .	8-18	creating . . . . .	8-2 to 8-10
ENDORSE . . . . .	8-18	database trait . . . . .	8-2
EVENT-DURATION . . . . .	8-19	defining . . . . .	12-6
EVENT-EFFECT . . . . .	8-19	defining print . . . . .	12-5
EVENT-STATUS . . . . .	8-19	description story . . . . .	8-3
EVENT-STYLE . . . . .	8-19	difference between forms and fonts . . . . .	12-5
LINE-COUNT . . . . .	8-20	introduction . . . . .	8-1, 8-2
LITERAL . . . . .	8-20	naming conventions . . . . .	8-2
MAIL-CC . . . . .	8-20	on and off setting . . . . .	12-6
MAIL-TO . . . . .	8-20	print queue . . . . .	8-30
MEDIA-ID . . . . .	8-20	printer profile . . . . .	12-14
MODIFY-BY . . . . .	8-20	queue . . . . .	8-1, 8-2, 8-3
MODIFY-DATE . . . . .	8-20	repeating a field . . . . .	8-13
MODIFY-DEV . . . . .	8-20	seek . . . . .	8-31, 14-62
NSML-LITERAL . . . . .	8-21	selecting . . . . .	12-26
PAGE-NUMBER . . . . .	8-21	selecting in style story . . . . .	12-26
PRESENTER . . . . .	8-21	setting up . . . . .	8-1 to 8-33
READY . . . . .	8-21	space limitations . . . . .	12-7
RESULT-INDEX . . . . .	8-21	timing . . . . .	8-29
RESULT-LOCATION . . . . .	8-22	transmitting . . . . .	14-89
SEARCH-ID . . . . .	8-22	wire story . . . . .	8-33
STATUS . . . . .	8-22	forms allowed . . . . .	5-29
STILL-ID . . . . .	8-22	free blocks . . . . .	20-18
STILL-PRESET . . . . .	8-22	free list	
TAPE-TIME . . . . .	8-22	add space from Dead queue . . . . .	A-14
TITLE . . . . .	8-23	high water mark . . . . .	11-22, 19-3
TOTAL-TIME . . . . .	8-23	introduction . . . . .	19-3
types defined . . . . .	8-13 to 8-24	low water mark . . . . .	11-23, 19-3
VAR-N . . . . .	8-23	rebuild . . . . .	A-16
VIDEO-ID . . . . .	8-23	size . . . . .	A-10
		free space . . . . .	20-18

- freezing of a console . . . . . 2-12
- FROM.NEWS.PACKAGES queue . . . . . 14-93
- FTS server . . . . . 14-67
- ftsidx.exe . . . . . 14-67
- ftssch.exe . . . . . 14-67
- fullform job list command . . . . . A-43
- function key console control command . . . . . 2-22
- function keys
- assigning a command to . . . . . 2-11
  - changing a command assignment . . . . . 2-11
  - deleting a command assignment . . . . . 2-11
  - displaying a command assignment . . . . . 2-12
  - extended macro keys . . . . . 7-24
  - macros . . . . . 7-6
  - predefined . . . . . 7-8
- G**
- general queue trait . . . . . 6-21
- grommet . . . . . 4-13
- group access . . . . . 6-20 to 6-28
- group assignments, transferring . . . . . 6-27
- group checker . . . . . 6-7
- avoiding circular reference . . . . . 6-16
  - deleting groups . . . . . 6-13
  - error messages . . . . . 6-9 to 6-12
  - recursion . . . . . 6-16
  - renamed groups . . . . . 6-13
- group names . . . . . 6-5
- group permissions . . . . . 6-21
- group security
- control system access . . . . . 6-20 to 6-28, 18-12
  - designing . . . . . 18-12
- group story queue . . . . . 4-29
- grouping footers for stories . . . . . 12-15
- grouping headers for stories . . . . . 12-15
- groups
- adding members . . . . . 6-15
  - as members . . . . . 6-15
  - assign to queues/directories . . . . . 6-22 to 6-27
  - circular references . . . . . 6-16
  - defining . . . . . 6-5
  - deleting . . . . . 6-13
  - managing from console . . . . . G-34
  - nesting . . . . . 6-15
  - recursion . . . . . 6-16
  - renaming . . . . . 6-12
  - specifying members . . . . . 6-6
  - transferring assignments . . . . . 6-27
- Groups queue . . . . . 6-3
- grpcheck . . . . . 6-7
- 8-bit characters . . . . . 14-82
  - console command . . . . . 4-29, A-20
- gtraits command
- deleting groups . . . . . 6-13
  - entering a group name . . . . . 6-6
  - interactive . . . . . 6-14
  - syntax . . . . . A-20
- gtraits i command . . . . . 6-14
- gtraits transfer command . . . . . 6-27
- H**
- halting print requests . . . . . 12-47
- handshake keyword . . . . . 13-20
- hard drive failure . . . . . 22-14
- headers
- grouping in queues . . . . . 12-15
  - printing . . . . . 12-15
  - user-selected . . . . . 12-16
- help console command . . . . . A-21
- HEOL (hard end of line) character . . . . . 13-17
- HEOL dialog command . . . . . A-51
- hidden categories, wire distribution . . . . . 13-35
- hiding commands . . . . . C-4
- high water mark
- checking . . . . . 11-20
  - definition . . . . . 11-22
  - viewing at console . . . . . 11-20
- highwater system profile parameter . . . . . 11-22

- history
    - console . . . . . 2-7 to 2-13
    - log names . . . . . 2-10
    - reading older . . . . . 2-10
    - viewing recent . . . . . 2-8
  - history command
    - bottom . . . . . 2-21
    - list. . . . . 2-22
    - print . . . . . 2-23
    - top . . . . . 2-23
    - up . . . . . 2-23
  - hogs command . . . . . 19-5, 19-6, A-21
  - hosts file. . . . . 11-17, 14-94
    - Rx and Tx links. . . . . 14-94
  - how to... *See procedures*
  - HTML
    - configuring. . . . . 15-3
    - sending via Tx/net. . . . . 15-2
    - story form. . . . . 15-5 to 15-23
      - contents. . . . . 15-20
      - sample. . . . . 15-18
- I**
- I/O ports
    - CCU II . . . . . D-8
    - CCU III. . . . . D-14
    - CCU IV. . . . . D-19
    - PCU . . . . . D-24
  - id system profile parameter. . . . . 11-22
  - idle keyword . . . . . 13-20
  - idle time-out . . . . . 6-19
  - idlecount, printer profile . . . . . 12-10
  - Ierrs, input errors. . . . . 22-16
  - ignore command . . . . . A-44
    - validation story . . . . . 14-28
  - ignore yes command . . . . . 14-30
  - ignore-del command . . . . . 14-12
  - ignore-del job list command . . . . . A-44
  - incorporating configuration changes. . . . . 11-10
  - increasing database space . . . . . 19-7
  - increasing your license allowance. . . . . 11-3
  - index . . . . . 5-29
  - index field . . . . . 5-28
    - blanked out . . . . . 5-31
    - compatibility errors . . . . . 5-30
    - default . . . . . 5-31
  - index value . . . . . 5-16
  - initprint, printer profile. . . . . 12-10
  - input errors (Ierrs). . . . . 22-16
  - insert char function . . . . . 7-24
  - install Fast Text Search (FTS). . . . . 14-66 to 14-74
  - interactive gtraits command . . . . . 6-14
  - intermediate queue, updating. . . . . 14-91
  - internationalization. . . . . 13-39
  - intersystem messaging . . . . . 11-51 to 11-57
  - invalid block, defined . . . . . 19-3
  - ISO character set . . . . . 13-30
  - ISO characters . . . . . 13-30
- J**
- job list
    - command set . . . . . 14-10
    - keyword search . . . . . 13-40
    - keyword server . . . . . 14-55
    - mailbox tasks . . . . . 14-13 to 14-18
    - multiple tasks. . . . . 14-23
    - network Tx link . . . . . 14-95
    - priority queue . . . . . 14-9
    - scan line . . . . . 14-9
    - suppressing a search . . . . . 13-42
    - tasks. . . . . 14-23, 14-38, 14-55
    - timed interval tasks . . . . . 14-18 to 14-21
    - validation. . . . . 14-29
  - job list command
    - at . . . . . A-42
    - bscan . . . . . A-42
    - distribution . . . . . A-43
    - dup. . . . . A-43

- job list command (continued)
- every . . . . . A-43
  - for Tx links . . . . . 14-97
  - fullform . . . . . A-43
  - ignore . . . . . A-44
  - ignore-del . . . . . 14-12, A-44
  - local . . . . . A-44
  - move . . . . . 14-11, A-44
  - number . . . . . A-44
  - on . . . . . A-45
  - open . . . . . 14-95, A-45
  - order . . . . . A-45
  - put . . . . . 14-96, A-45
  - quiet . . . . . A-46
  - remote . . . . . A-46
  - remove . . . . . 14-96, A-46
  - replace . . . . . A-46
  - scan . . . . . A-46
  - send-del . . . . . 14-12, A-47
  - source . . . . . A-47
  - validate . . . . . 14-30, A-47
- join option parameter . . . . . 13-4
- K**
- keyboard check messages for macros . . . . . C-11
  - keyboard check program messages . . . . . C-11
  - keyboard description story
    - default PAUSE value . . . . . 11-26
  - keyboard macros
    - assigning . . . . . 7-16
    - dictionary . . . . . C-40
    - key names . . . . . 7-7
  - keyboard stories
    - error checking . . . . . 7-13
    - repeating macros for VT . . . . . 7-20
  - keyboards
    - creating . . . . . 7-3
    - macros . . . . . 7-5
    - customizing . . . . . 7-3
  - macro segments . . . . . 7-5
    - switching . . . . . 7-26
  - VT/DOS terminals . . . . . 7-19 to 7-26
  - keyword search
    - creating search rules . . . . . 13-41
    - default directory paths . . . . . 13-48
    - default job list entry . . . . . 13-42
    - destination . . . . . 13-44
    - job list . . . . . 13-40
    - keyword story . . . . . 13-41
    - keyword story queue . . . . . 13-40
    - kwd mailbox . . . . . 13-48
    - matching . . . . . 13-48
    - maximum keywords . . . . . 13-43
    - personal queues . . . . . 13-44, 14-57
    - purge interval . . . . . 13-44, 14-57
    - removing a rule set . . . . . 13-47
    - rule sets . . . . . 13-41
    - search rules . . . . . 13-44, 14-57
    - setting up . . . . . 13-39
    - suppress searching . . . . . 13-42
    - user notification . . . . . 13-47
    - wires . . . . . 13-39 to 13-50
  - keyword servers
    - adding . . . . . 14-51 to 14-60
    - defined . . . . . 14-2
    - job list . . . . . 14-55
    - mailbox . . . . . 14-17
    - managing . . . . . 14-51
    - performance implications . . . . . 14-51
    - search job list . . . . . 14-58
    - search rules . . . . . 14-56
    - suppress searching . . . . . 13-42
  - keyword story
    - creating . . . . . 13-41
    - saving . . . . . 13-42
  - keywords
    - accent . . . . . 13-13
    - answer . . . . . 13-13
    - bell . . . . . 13-13
    - bits . . . . . 13-13

- keywords (continued)
- category . . . . . 13-15
  - distribution. . . . . 13-15
  - end . . . . . 13-16
  - eol. . . . . 13-16
  - eop . . . . . 13-17
  - eos . . . . . 13-17
  - flags . . . . . 13-18
  - flash . . . . . 13-19
  - form . . . . . 13-20
  - handshake . . . . . 13-20
  - idle . . . . . 13-20
  - map . . . . . 13-21
  - pagenumber . . . . . 13-22
  - paragraph. . . . . 13-22
  - parity . . . . . 13-22
  - presenter. . . . . 13-23
  - priority . . . . . 13-23
  - reverse . . . . . 13-24
  - start . . . . . 13-24
  - stopbits. . . . . 13-25
  - title . . . . . 13-25
  - tli (temp). . . . . 13-26
- keywords for wire profiles. . . . . 13-13 to 13-26
- killed a story, who last. . . . . 5-17
- killed stories
- Dead queue . . . . . 5-19
  - recovering . . . . . 5-20
- kwd mailbox . . . . . 13-48
- L**
- last login option. . . . . 18-8
  - lastlogin system profile parameter . . . . . 11-22
  - LHDM-WObfpRmF, explained . . . . . 5-17
  - license allowance, increasing . . . . . 11-3
  - license limitations/categories . . . . . 11-2
  - line formatting, heol command . . . . . A-53
  - LINE-COUNT form field . . . . . 8-20
  - linking queues to a server . . . . . 14-15
  - links to other iNews products . . . . . 1-6
  - list c command
    - CCU parameter . . . . . 11-6
    - COMPUTER parameter . . . . . 11-6
    - DEV parameter . . . . . 11-5
    - DEVICE\_TYPE parameter . . . . . 11-6
    - DEVNAME parameter. . . . . 11-7
    - OPTIONS parameter . . . . . 11-7
    - PRINTER parameter . . . . . 11-7
    - SPEED parameter. . . . . 11-7
    - to view device information . . . . . 11-4
  - list console control command . . . . . 2-22
    - send output to printer . . . . . 5-15
  - list d command
    - introduction. . . . . 5-48
    - syntax. . . . . A-23
  - list d-g . . . . . 6-3
  - list g command, syntax . . . . . A-23
  - list q command . . . . . A-24
    - story information. . . . . 5-15
    - syntax. . . . . A-24
  - list qindex . . . . . 5-17
  - list s command . . . . . 4-37, A-24
  - list u command . . . . . 22-2, A-25, G-10
    - syntax. . . . . A-25
  - list u-v command, user traits. . . . . G-2
  - listing a file . . . . . 11-10
  - listing all computers . . . . . A-6
  - listing contents of a tape . . . . . 20-6
  - listing last password change . . . . . G-7
  - listing parameter settings . . . . . 11-20
  - listing password change by date . . . . . G-8 to G-10
  - listing tape contents. . . . . 20-11
  - listing users logged in . . . . . 18-9
  - LITERAL form field . . . . . 8-20
  - load balancing. . . . . 11-22
  - load keyboard command. . . . . 7-26
  - load system profile parameter. . . . . 11-22
  - loading tapes. . . . . 20-2
  - local job list command. . . . . A-44
  - local print, style options . . . . . 12-38



local printing . . . . . 12-31 to 12-46  
     using styles for VT . . . . . 12-30  
 localtimeout parameter, changing . . . . . 11-19  
 localtimeout system profile parameter . . . . . 11-23  
 locked blocks, checking for. . . . . 22-11  
 locked queues, who last . . . . . 5-48  
 locked stories, removing. . . . . A-11  
 locking queues, group permissions . . . . . 6-21  
 locks  
     edit . . . . . 5-50  
     explained. . . . . 5-50  
     order . . . . . 5-52  
     production . . . . . 5-53  
     user . . . . . 5-51  
 lockuser database trait . . . . . 5-48  
 log history names . . . . . 2-10  
 logclose console control command. . . . . 2-22  
 logging in  
     as superuser . . . . . 3-3  
     on the console . . . . . 3-2  
     users unable to . . . . . 22-2  
 logging out  
     automatic . . . . . 6-19  
     from a remote console. . . . . 2-16  
     remote users . . . . . 2-16  
 logical printer number . . . . . 11-43  
 login time-out. . . . . 6-20  
 logins, recording. . . . . 18-10  
 logintimeout system profile parameter . . . . . 11-23  
 logopen console control command. . . . . 2-22  
 logout command . . . . . 2-22, A-26  
 low on space message. . . . . 19-4  
 low water mark  
     checking . . . . . 11-20  
     free list. . . . . 19-3  
     parameter definition . . . . . 11-23  
     viewing at console. . . . . 11-20  
 lowercase in passwords . . . . . 3-3  
 lowwater system profile parameter . . . . . 11-23  
 ls console command . . . . . A-27

## M

machines category, defined . . . . . 11-3  
 macros  
     adding comments . . . . . 7-6  
     creating. . . . . 7-5  
     defined . . . . . 7-5  
     for function keys. . . . . 7-6  
     key combinations . . . . . 7-10  
     plain text in. . . . . 7-11  
     predefined function keys . . . . . 7-8  
     repeating. . . . . 7-11, 7-20  
     segments. . . . . 7-5  
     using state keys. . . . . 7-9  
 VT  
     assigning to standard keys . . . . . 7-22  
     blank command. . . . . 7-20  
     extended macro keys. . . . . 7-24  
     extended vs. standard keys . . . . . 7-23  
     pause command . . . . . 7-20  
     Pause/Break key. . . . . 7-25  
     repeating. . . . . 7-20  
 mail  
     8-bit characters . . . . . 14-82  
     character conversion table . . . . . 14-83  
     creating an alias . . . . . 6-29  
     dead letter queue . . . . . 5-8  
     disabling to all users. . . . . 14-81  
     network mail . . . . . 14-81  
     outgoing mail queue. . . . . 5-7  
     sending to another system . . . . . 14-81  
     SYSTEM.MAIL.ERROR . . . . . 5-8  
 mail aliases . . . . . 6-28 to 6-31  
     creating. . . . . 6-29  
     over the network. . . . . 6-30  
 Mail Compose strings . . . . . 14-83  
 mail server messages . . . . . C-13  
 mail servers . . . . . 14-80 to 14-88  
 mail story . . . . . 8-29  
 mail to groups . . . . . 6-28 to 6-31

- mailbox tasks . . . . . 14-13, 14-95
  - timed interval tasks . . . . . 14-14, 14-19
- mailboxes
  - all . . . . . 14-16
  - assigning to queues . . . . . 14-17, G-31
  - debugging display . . . . . 11-25
  - defined . . . . . 14-15
  - dist (distribution) . . . . . 13-38, 14-16
  - grp (group) . . . . . 14-17
  - key (keyboard) . . . . . 14-17
  - kwd (keyword) . . . . . 13-43, 13-48, 14-17
  - listing all queues (at console) . . . . . G-31
  - number available . . . . . 14-14
  - reserved . . . . . 14-16
  - reserved, defined . . . . . 14-14
  - setting up . . . . . 14-14
  - standard . . . . . 14-17
  - standard, defined . . . . . 14-14
  - wire distribution story . . . . . 13-38
- MAIL-CC form field . . . . . 8-20
- mailllookup . . . . . F-7
- MAIL-TO form field . . . . . 8-20
- maintaining database storage . . . . . 19-8 to 19-13
- makeccutab
  - VT character mapping . . . . . C-46
- makeccutab console command . . . . . A-28
- makeshift command . . . . . C-45
- maketab console command . . . . . A-29
- making tape backups . . . . . A-35
- map dialog command . . . . . A-51
- map keyword . . . . . 13-21
- map, printer profile . . . . . 12-11
- mapin dialog command, definition . . . . . A-52
- mapout dialog command . . . . . A-52
- mapping characters . . . . . 12-11
  - Netstation to Avstar NRCS . . . . . 8-34
- master computer
  - designating . . . . . 11-24
  - putting back online . . . . . 21-16
- master phone list, updating . . . . . 14-21
- master system profile parameter . . . . . 11-24
- matching and case in a distribution story . . . . . 14-36
- matching and order in a distribution story . . . . . 14-36
- maxhits system profile parameter . . . . . 11-24
- Media Browse system, described . . . . . 1-5
- MEDIA-ID form field . . . . . 8-20
- members of a group . . . . . 6-6
- membership list
  - adding groups . . . . . 6-15
  - circular references . . . . . 6-16
  - recursion . . . . . 6-16
- merging multiple takes . . . . . 13-4
- Message bar button . . . . . 6-7
- message dialog command . . . . . 17-3, A-53
- message field . . . . . 6-7
- messages, broadcast . . . . . A-4
- MIME . . . . . 14-82
- min\_passwd\_length system profile
  - parameter . . . . . 11-24
- minimum password length . . . . . 11-24
- minus four option . . . . . 22-8
- modem console control command . . . . . 2-22
- modems
  - dial dictionary . . . . . C-39
  - resetting for a wire . . . . . 22-8
- modify user account dialog box . . . . . 4-4 to 4-9
- MODIFY-BY form field . . . . . 8-20
- MODIFY-DATE form field . . . . . 8-20
- MODIFY-DEV field . . . . . 18-2
- MODIFY-DEV form field . . . . . 8-20
- modifying
  - from the console . . . . . G-3
  - multiple groups . . . . . 6-14
  - system profiles . . . . . 11-19
  - user traits/accounts . . . . . 4-3
- modifying user traits . . . . . 4-3 to 4-4
- monitoring free space . . . . . 19-2 to 19-5
- monitoring the free list . . . . . 19-3
- move command . . . . . 14-11
  - assigning distribution codes . . . . . 14-34
- move job list command . . . . . A-44
- moved a story, who last . . . . . 5-17

moving a printer . . . . . 11-41  
 msgclean console command . . . . . A-30  
 msgmailalert . . . . . F-8  
 msgserver system profile parameter . . . . . 11-25  
 multiple queues . . . . . 14-91  
   updating intermediate queues . . . . . 14-92  
 multiple servers . . . . . 14-88

## N

name system profile parameter . . . . . 11-25  
 naming conventions for forms . . . . . 8-2  
 naming groups . . . . . 6-5, 6-12  
 nesting groups . . . . . 6-15  
 net system profile parameter . . . . . 11-25  
 netlogintimeout  
   system profile parameter . . . . . 11-25  
 netstat -i command . . . . . 22-15  
 Netstation, mapping characters . . . . . 8-34  
 network CCU Ethernet . . . . . D-3  
 network failure . . . . . 22-14  
 network link, device numbering . . . . . 14-94  
 network mail . . . . . 14-81  
 network resource, adding to /site/config . . . . . 14-98  
 network Rx link  
   DECnet . . . . . 14-94  
   introduction . . . . . 14-88, 14-93  
 network services . . . . . 17-6  
   adding SCO UNIX . . . . . 17-8  
   adding SGI IRIX . . . . . 17-8  
 network Tx link  
   configuration . . . . . 14-98  
   DECnet . . . . . 14-94  
   introduction . . . . . 14-93  
   job list . . . . . 14-95  
   security . . . . . 14-96  
   transferring stories . . . . . 14-88  
 networks, device type . . . . . 11-3  
 news database, overview . . . . . 5-1 to 5-13  
 nonprinting characters . . . . . 12-19, 13-26  
   alias . . . . . 13-26  
   decimal value . . . . . 13-27  
 nonspacing diacritical mark . . . . . 13-28  
 NOT search rule . . . . . 13-45  
 notification priority . . . . . 13-33  
 notification, group . . . . . 6-25  
 notify group . . . . . 5-35  
 NS directory . . . . . 11-40  
 ns.exe . . . . . 11-40  
 NSML . . . . . 15-10  
 NSML-LITERAL form field . . . . . 8-21  
 number command . . . . . 14-12, A-30  
 number job list command . . . . . A-44  
 number of blocks, list . . . . . A-8  
 number of printers on system . . . . . 11-42

## O

Oerrs, output errors . . . . . 22-15  
 offline command . . . . . A-31  
   defined . . . . . 3-10  
 offline printer . . . . . 12-48  
 on and off setting, forms . . . . . 12-6  
 on command . . . . . 14-19  
 on job list command . . . . . A-45  
 online command . . . . . 5-13, A-31  
 open & put commands . . . . . 14-13  
 open job list command . . . . . 14-95, A-45  
 optional format strings . . . . . 15-6, 15-14  
   time and date elements . . . . . 15-15  
   time and date fields . . . . . 15-17  
 OPTIONS parameter, list c command . . . . . 11-7  
 options, for wire profiles . . . . . 13-13 to 13-26  
 OR search rule . . . . . 13-45  
 order command . . . . . 14-13  
 order job list command . . . . . A-45  
 order lock . . . . . 5-52  
   unbusyng . . . . . 22-6  
 ordered checkbox . . . . . 5-30

ordered queue, who last . . . . . 5-48  
 ordering a pending print request . . . . . 12-48  
 ordering queues . . . . . 5-33  
   changing . . . . . 14-92  
 orderuser database trait . . . . . 5-48  
 otod console command . . . . . A-31  
 out-of-space condition . . . . . 19-5  
 output errors (Oerrs) . . . . . 22-15  
 outtime, default . . . . . 11-26  
   system profile parameter . . . . . 11-26

## P

pagefooter, printer profile . . . . . 12-14, 12-16  
 pageheader, printer profile . . . . . 12-14, 12-16  
 pagelength, printer profile . . . . . 12-14  
 PAGE-NUMBER form field . . . . . 8-21  
 pagenumber keyword . . . . . 13-22  
 paragraph keyword . . . . . 13-22  
 parallel wire servers  
   defined . . . . . 14-2  
   illustrated . . . . . 14-41  
   installing . . . . . 14-43 to 14-51  
   managing . . . . . 14-41 to 14-42  
   queues and traits . . . . . 14-42, 14-52  
 parameter list, system profile . . . . . 11-21  
 parameter settings, listing . . . . . 11-20  
 parentheses in a search . . . . . 13-45  
 parity keyword . . . . . 13-22  
 pass dialog command . . . . . A-53  
 passwd root command . . . . . 3-4  
 passwords  
   changing superuser . . . . . 3-4  
   changing system operator . . . . . 3-4  
   changing user . . . . . 4-9  
   checking . . . . . G-10  
   checking for users without . . . . . 18-3, G-7  
   forcing change . . . . . 18-6  
   listing change by date . . . . . G-8 to G-10  
   listing last change . . . . . 18-6, G-7

  minimum length . . . . . 11-24  
   required minimum length . . . . . 18-2  
   restrictions . . . . . 18-7  
   security procedures . . . . . 18-2  
   setting and changing for users . . . . . 18-3  
   status . . . . . 18-3  
   system administration . . . . . 3-3 to 3-5  
   using upper and lowercase . . . . . 3-3  
 pause command . . . . . A-54  
   for VT . . . . . 7-20  
 Pause/Break key . . . . . 7-25  
 pausetimeout system profile parameter . . . . . 11-26  
 pausing screen display . . . . . 2-8  
 PCU  
   back panel . . . . . D-24  
   front panel . . . . . D-22  
   I/O ports . . . . . D-24  
   introduction . . . . . D-2  
   LED codes . . . . . D-22  
   LED display . . . . . D-22  
   reset switch . . . . . D-21  
 pending print request . . . . . 12-47  
 People directory . . . . . 4-23  
 period with a command, using . . . . . 3-8  
 Peripheral . . . . . D-2  
 peripheral controller unit *See PCU*  
 permissions  
   assigning read and write . . . . . 6-22 to 6-27  
 PHONELISTS.ALL . . . . . 14-15  
 picolor . . . . . F-10  
 ping, testing connectivity . . . . . 22-15  
 pipe symbol (|) . . . . . 22-12  
 piping with fgrep . . . . . 22-12  
 plain text in macros . . . . . 7-11  
 platform parameter . . . . . 11-36  
 pointers in a story . . . . . 19-4  
 polarity . . . . . 22-9  
 policies for backups . . . . . 20-2  
 power failure . . . . . 22-13  
 PRESENTER form field . . . . . 8-21  
 presenter keyword . . . . . 13-23

- preview lines in Queue panel . . . . . 4-17
- print command . . . . . 2-23, A-32
  - examples . . . . . 10-7
- print effects, combining . . . . . 12-4
- print forms . . . . . 12-5
- print jobs
  - cancelling . . . . . 12-48
  - runaway . . . . . 12-48
- print quality, selecting . . . . . 12-5
- print queue . . . . . 12-47
  - forms . . . . . 8-30
  - rearranging . . . . . 12-47
  - removing pending request . . . . . 12-47
- print request . . . . . 8-30
  - aborting . . . . . 12-47
  - current . . . . . 12-47
  - ordering . . . . . 12-48
  - pending . . . . . 12-47
  - rearranging . . . . . 12-47
  - removing pending . . . . . 12-47
  - restarting current . . . . . 12-48
- print script command, expanded print . . . 12-10
- print servers . . . . . 14-78
  - adding . . . . . 14-79 to 14-80
  - configuration . . . . . 14-79
  - configuration line . . . . . 14-80
  - printer profile . . . . . 14-80
- print styles . . . . . 5-36, 12-23
- printer, send output from list command . . . 5-15
- printer # parameter . . . . . 11-36, 11-43
- printer control code
  - fonts . . . . . 12-4
  - unusable characters . . . . . 12-20
- PRINTER display, console . . . . . 2-22
- printer offline response . . . . . 12-48
- PRINTER parameter, list c command . . . . 11-7
- printer profile
  - adding characters . . . . . 12-19
  - autocue option . . . . . 12-8
  - changing options . . . . . 12-25
  - characters used by the system . . . . . 12-20
  - combining forms . . . . . 12-6
  - combining print effects . . . . . 12-4
  - computer copies . . . . . 11-44
  - copying a standard profile . . . . . 11-44
  - corresponding to printer ports . . . . . 22-10
  - creating . . . . . 11-44
  - file numbering . . . . . 12-2
  - font and form space . . . . . 12-7
  - footers . . . . . 12-15
  - forms . . . . . 12-5
  - including angle brackets . . . . . 12-20
  - listing directory . . . . . 12-2
  - nonprinting character . . . . . 12-19 to 12-23
  - option defaults . . . . . 12-17
  - options . . . . . 12-3, 12-7 to 12-19
  - pagefooter . . . . . 12-16
  - pageheader . . . . . 12-16
  - print servers . . . . . 14-80
  - profile and style options . . . . . 12-13
  - profile only options . . . . . 12-8
  - scriptrhmax . . . . . 12-14
  - special characters . . . . . 12-19 to 12-23
  - translating characters . . . . . 12-11
  - user-selected headers/footers . . . . . 12-16
- printer queue . . . . . 11-42
- printers
  - adding . . . . . 11-41 to 11-47
  - changing CCUs . . . . . 11-42
  - combining print effects . . . . . 12-4
  - configuration . . . . . 11-42
  - customizing operations . . . . . 12-1, 13-1
  - default options . . . . . 12-7
  - device number . . . . . 11-41
  - device types . . . . . 11-3
  - expanded text mode . . . . . 12-10
  - listing files . . . . . 12-2
  - local print on VT . . . . . 12-30
  - local print styles . . . . . 12-38
  - logical printer number . . . . . 11-43
  - mapping characters . . . . . 12-11
  - maximum system capacity . . . . . 11-42

- printers (continued)
  - moving . . . . . 11-41
  - offline . . . . . 12-48
  - printer queue . . . . . 11-42
  - problems . . . . . 22-9
  - selecting printer output . . . . . 12-26
  - setting baud rate. . . . . 11-43
- printing
  - aborting print requests . . . . . 12-47
  - characters not on the keyboard . . . . . 12-11
  - deleting runaway print jobs . . . . . 22-10
  - operations. . . . . 12-47
  - pending print requests . . . . . 12-47
  - removing pending print requests. . . . . 12-47
  - uses for abstract . . . . . 5-37
- priority keyword . . . . . 13-23
- priority notification, disabling. . . . . 13-34
- priority queue . . . . . 14-9
- problems
  - with printers. . . . . 22-9
  - with wires . . . . . 22-7
- procedures
  - access Registry Editor. . . . . F-2
  - assign function key command . . . . . 2-11
  - back up site files to tape . . . . . 20-25
  - cancel runaway print jobs . . . . . 12-48
  - capture bitmaps . . . . . 9-8 to 9-10
  - change closed captioning text color . . . . . F-3
  - change presenter instructions text color. F-10
  - changing database traits. . . . . 5-21 to 5-24
  - changing user passwords. . . . . 4-9
  - changing user preferences . . . . . 4-10
  - copying user accounts . . . . . 4-26
  - create emergency boot/root disks . . . . . 20-24
  - create new CG template . . . . . 9-16 to 9-20
  - create new form . . . . . 8-3
  - create new story . . . . . 5-9
  - create new user account . . . . . 4-27
  - creating database manager (dbmanager)
    - account . . . . . 4-36
  - creating directories. . . 4-23 to 4-24, 5-4 to 5-6
  - creating queues . . . . 4-24 to 4-25, 5-6 to 5-7
  - creating user manager (umanager)
    - account . . . . . 4-35
  - define key used to advance timing bar . F-12
  - define local print banner style . . . . . 12-44
  - define variable for setting story form field text
    - limit . . . . . F-18
  - display function key assignment . . . . . 2-12
  - editing a story . . . . . 13-8
  - enable destination order . . . . . F-5
  - enable message mail alerts. . . . . F-8
  - enable synchronized timing. . . . . F-17
  - exit console program . . . . . 2-13, 2-21
  - find who moved/dup/killed story . . . . . 5-17
  - hide groups from e-mail lists. . . . . F-7
  - install capture tool . . . . . 9-8
  - install FTS programs . . . . . 14-67
  - launching CG template editor . . . . . 9-11
  - launching UNIX line editor (ed) . . . . . 10-5
  - list group information . . . . . 6-3
  - log in as system operator . . . . . 3-2
  - log out all users . . . . . 4-37
  - log out remote users . . . . . 2-16
  - modify user traits. . . . . 4-3
  - pause screen display . . . . . 2-8
  - prevent access to CG title entry . . . . . 9-24
  - recovering killed stories. . . . . 5-20
  - remove directory or queue. . . . . 5-10
  - remove locks at startup . . . . . 3-8
  - remove print request . . . . . 12-47
  - remove user accounts . . . . . 4-33
  - renaming directories and queue . . . . . 5-10
  - restart print request . . . . . 12-48
  - restarting queue sort function . . . . . 5-33
  - search user's last login. . . . . 18-8
  - searching text with ed, UNIX line editor. 10-7
  - select A and B servers . . . . . 2-5
  - select all servers at console. . . . . 2-6
  - select one server. . . . . 2-5
  - set automatic update . . . . . 14-90
  - shut down system . . . . . 3-10

- procedures (continued)
- simplified user limits . . . . . 4-20
  - start or restart queue sorting . . . . . G-27
  - to unfreeze console . . . . . 2-12
  - unbusy (unlock) story or queue . . . . . 5-53
  - unlock queue with unknown key . . . . . 5-52
  - unlocking stories without keys . . . . . 5-51
  - view console history . . . . . 2-9
  - view console's configuration file. . . . . 2-17
  - viewing database traits . . . . . 5-13 to 5-21
  - zoom in on one server. . . . . 2-7
- processes status . . . . . 22-12
- processing deleted stories. . . . . 14-12
- production cue set story entity . . . . . 15-5
- production cue, example . . . . . 9-2
- profile and style options . . . . . 12-13
- profile options . . . . . 12-25
- defaults . . . . . 12-17
- profile-only options . . . . . 12-8
- program parameter . . . . . 11-36
- programs invoked by Avstar NRCS . . . . . A-2
- prompt, superuser . . . . . 3-3
- ps command . . . . . 22-11
- punctuation, in the configuration file . . . . . 11-10
- purge interval
- choosing . . . . . 5-41
  - database trait. . . . . G-27
  - defined . . . . . 19-2
  - introduction . . . . . 19-2
  - length . . . . . 5-41
  - purge limit . . . . . 5-42
  - reducing . . . . . 5-41
  - setting . . . . . G-28
  - wire distribution story . . . . . 13-38
- purge limit
- changing . . . . . 5-42
  - guidelines . . . . . 11-26
  - setting . . . . . 11-26
  - viewing at console. . . . . 11-20
- purgelimit system profile parameter . . . . . 11-26
- purging stories . . . . . 19-2
- put and open commands . . . . . 14-13
- put job list command . . . . . 14-96, A-45
- ## Q
- Q.SEEK. . . . . C-27
- QIC tapes . . . . . 20-2
- queue form. . . . . 5-27
- Queue panel, preview lines . . . . . 4-17
- queues
- apply changes to all . . . . . 5-24
  - applying multiple permissions. . . . . 6-26
  - assigning a form to . . . . . 8-11
  - automatic refresh . . . . . G-34
  - backing up . . . . . 20-3, 20-7, 20-20
  - choosing a purge interval . . . . . 5-41
  - choosing queue to purge . . . . . 5-40
  - creating a dialog . . . . . 17-3
  - dictionary . . . . . C-27
  - forms. . . . . 8-1, 8-2
  - hiding . . . . . 6-27
  - linking to a server. . . . . 14-15
  - matching purge intervals . . . . . 5-41
  - priority . . . . . 14-9
  - purge limit and dbserver . . . . . 11-26
  - removing . . . . . 5-10
  - renaming . . . . . 5-10
  - restart sort from console . . . . . G-26
  - restart sort/ordering. . . . . 5-33
  - security . . . . . 6-18
  - skip flags. . . . . 20-7
  - sort . . . . . A-14
  - transmitting order changes. . . . . 14-92
  - update. . . . . 14-90
  - updating intermediate queues . . . . . 14-91
  - who last locked. . . . . 5-48
  - who last ordered. . . . . 5-48
- quick index value, one-letter flags . . 5-16 to 5-17
- quiet job list command. . . . . A-46
- quiet no command . . . . . 14-30

## R

- r option . . . . . 5-12
- read group . . . . . 5-34
  - assigning readers . . . . . 6-23
  - setting traits for groups . . . . . 6-23
- read only option, CGs . . . . . 9-19
- read rates . . . . . 4-7
  - default . . . . . 11-27
- reading and writing, restricting . . . . . 6-26
- reading older history . . . . . 2-10
- readrate system profile parameter . . . . . 11-27
- READY form field . . . . . 8-21
- rearranging print requests . . . . . 12-47
- receive link *See Rx links*
- receiving mail . . . . . 4-29
- recent console history . . . . . 2-8
- reclaiming Dead queue space . . . . . 19-6
- reconnect command . . . . . A-33
- recording logins . . . . . 18-10
- recovering a failed system . . . . . 21-9
- recovering a killed story . . . . . 5-20
- recovering a story
  - by word . . . . . 20-15
  - maximum . . . . . 20-17
- recursion in groups . . . . . 6-16
- red data, polarity . . . . . 22-9
- refresh database trait . . . . . G-34
- relative date, defined . . . . . G-9
- remote console . . . . . 2-14 to 2-16
  - logging out from . . . . . 2-16
- remote job list command . . . . . A-46
- remote user
  - logging out from the main console . . . . . 2-16
- remote workstation, idle time-out . . . . . 11-27
- remotetimeout system profile parameter . . . . . 11-27
- remove command . . . . . 14-11
- remove job list command . . . . . 14-96, A-46
- removing
  - a directory or queue . . . . . 5-10
  - directory security from console . . . . . G-37
  - locks . . . . . A-7
  - queue security from console . . . . . G-37
  - stories . . . . . 14-96, A-11
  - user accounts . . . . . 4-33 to 4-34
- rename command . . . . . 5-10, 5-11, A-33
  - r option . . . . . 5-12
- renaming a group . . . . . 6-12
- renaming directories and queues . . . . . 5-10
- reordering a pending print request . . . . . 12-48
- repeating macros . . . . . 7-11, 7-20
  - duplicate or kill commands . . . . . 7-22
- replace command . . . . . 14-11
- replace job list command . . . . . A-46
- reserved mailboxes . . . . . 14-16
- reset console control command . . . . . 2-23
- resetting a wire's modem . . . . . 22-8
- resources category
  - defined . . . . . 11-3
- restart all command . . . . . 5-13
- restart console command . . . . . A-34
- restarting
  - current print request . . . . . 12-48
- restoring dictionary defaults . . . . . C-8
- restoring from tape
  - checking space available . . . . . 20-18
  - first level directory . . . . . 20-10
  - list date of each backup . . . . . 20-14
  - listing tape backup dates . . . . . 20-11, 20-13
  - listing tape contents . . . . . 20-11
- restricting access to queues & folders . . . . . 6-27
- restricting reading and writing . . . . . 6-26
- restricting security, system profile . . . . . 6-18
- RESULT-INDEX form field . . . . . 8-21
- RESULT-LOCATION form field . . . . . 8-22
- reverse keyword . . . . . 13-24
- RFC 2045 . . . . . 14-82
- RGB hexadecimal color chart . . . . . F-11
- rmdir console command . . . . . A-34
- rotating tape dumps . . . . . 20-2
- rr kb su m SOEKCVT sc, explained . . . . . A-25



- rule set
    - destination line . . . . . 13-44
    - introduction . . . . . 13-41
    - removing . . . . . 13-47
  - runaway print jobs . . . . . 12-48
    - eliminating . . . . . 22-10
  - rundown queue
    - display . . . . . G-33
    - setting display lines . . . . . G-32
  - Rx link . . . . . 14-95, 14-99
    - adding . . . . . 14-93
    - automatic story retrieval . . . . . 14-93 to 14-100
    - avoiding duplicate stories . . . . . 14-90
    - introduction . . . . . 14-88
    - network . . . . . 14-88, 14-93 to 14-100
    - serial . . . . . 14-88
  - Rx/Tx links, device type . . . . . 11-3
- S**
- scan codes . . . . . F-13
  - scan job list command . . . . . A-46
  - scan line . . . . . 14-9
  - scanning queues . . . . . A-7
  - SCO servers, shutting down . . . . . 3-12
  - SCO UNIX system
    - adding network services . . . . . 17-8
    - startup . . . . . 3-6
  - screen display, pausing . . . . . 2-8
  - SCRIPT command
    - default left margin . . . . . 11-27
    - default width . . . . . 11-27
  - scriptform, printer profile . . . . . 12-14
  - scriptlmax system profile parameter . . . . . 11-27
  - scriptlmax, printer profile . . . . . 12-14
  - scriptrhmax system profile parameter . . . . . 11-28
  - scriptrhstart, printer profile . . . . . 12-15
  - scriptshift, printer profile . . . . . 12-13
  - scriptspacing, printer profile . . . . . 12-15
  - search job list, keyword server . . . . . 14-58
  - search rules
    - AND . . . . . 13-45
    - creating . . . . . 13-41
    - evaluation order . . . . . 13-45
    - format . . . . . 13-41
    - keyword search . . . . . 13-44, 14-57
    - keyword servers . . . . . 14-56
    - length . . . . . 13-46
    - matching . . . . . 13-45, 13-46
    - multibyte characters . . . . . 13-44, 14-57
    - multiple words . . . . . 13-46
    - NOT . . . . . 13-45
    - OR . . . . . 13-45
    - parentheses . . . . . 13-45
    - rule sets . . . . . 13-41
  - searches, suppressing . . . . . 13-42
  - SEARCH-ID form field . . . . . 8-22
  - searching
    - by word and date . . . . . 20-16
    - for stories by words . . . . . 20-15
    - for stories, word and day . . . . . 20-17
    - for stories, word and month . . . . . 20-17
    - specifying number of stories . . . . . 20-17
  - searching a tape . . . . . 20-14
    - by date . . . . . 20-16
    - by word . . . . . 20-15
  - searching user accounts . . . . . 4-29 to 4-33
  - searchtape command . . . . . 20-15
  - searchtape console command . . . . . A-35
  - searchtape notes . . . . . 20-19
  - second (alternate) host section . . . . . 11-14
  - security
    - apply multiple permissions to queues . . . . . 6-26
    - assigning groups . . . . . 6-22 to 6-27
    - automatic workstation log out . . . . . 6-19
    - blacklist . . . . . 14-96
    - CG title entry . . . . . 9-23
    - change directories and queues . . . . . 5-21
    - effect of general queue trait . . . . . 6-21
    - groups . . . . . 18-12
    - hiding a queue . . . . . 6-27

- security (continued)
  - idle time-out . . . . . 6-19
  - levels, combining . . . . . 6-17
  - login time-out . . . . . 6-20
  - managing user accounts . . . . . 4-35
  - modem . . . . . 18-2
  - protecting SYSTEM directory . . . . . 6-28
    - from console . . . . . G-36
  - removing locks . . . . . 5-51
- security procedures . . . . . 18-2 to 18-3
  - MODIFY-DEV field . . . . . 18-2
  - passwords . . . . . 18-2
  - security modem . . . . . 18-2
  - superuser status . . . . . 18-2
- security system profile parameter . . 6-18, 11-28
- SEEK command
  - maxhits value . . . . . 11-24
  - seek form . . . . . 8-31
- seek form . . . . . 14-62
  - introduction . . . . . 14-62
- seek form field
  - by . . . . . 8-32
  - hits so far . . . . . 8-32
  - max hits . . . . . 8-32
  - notify hits . . . . . 8-32
  - results queue . . . . . 8-32
  - search for . . . . . 8-32
  - search type . . . . . 8-32
  - search where . . . . . 8-32
  - started at . . . . . 8-32
  - status . . . . . 8-32
- seek queue
  - assigning a form . . . . . 14-62
  - creating . . . . . 14-62
  - mailbox . . . . . 14-62
- seek servers
  - adding . . . . . 14-61 to 14-63
  - choosing a form . . . . . 14-62
  - configuration . . . . . 14-61
  - defined . . . . . 14-2
  - installing . . . . . 14-61
  - messages . . . . . C-14
  - selecting a server . . . . . 2-5 to 2-7
- semicolons
  - console prompt . . . . . 3-3
  - to precede comments . . . . . 11-10
  - to separate sections with blank lines . . 11-10
- send console command . . . . . A-35
- send-del command . . . . . 14-12
- send-del job list command . . . . . A-47
- sending HTML . . . . . 15-2
- separator, printer profile . . . . . 12-15
- serial Rx link, sending stories . . . . . 14-88
- serial Tx link, introduction . . . . . 14-88
- server commands, described . . . . . 2-3
- server program messages . . . . . C-9
- servers
  - adding . . . . . 14-22
  - connecting multiple . . . . . 14-88
  - defined . . . . . 11-3, 14-1
  - deleting stories . . . . . 14-11
  - duplicating stories . . . . . 14-10
  - mailbox tasks . . . . . 14-13
  - multiple tasks . . . . . 14-23
  - numbering stories . . . . . 14-12
  - parallel wire servers . . . . . 14-41
  - priority queue . . . . . 14-9
  - processing deleted stories . . . . . 14-12
  - tasks . . . . . 14-23, 14-38, 14-55
  - timed interval tasks . . . . . 14-18
  - using validation . . . . . 14-27
- service line, adding . . . . . 17-6
- service table
  - adding a network service . . . . . 17-6
  - adding a service . . . . . 17-7
  - line format . . . . . 17-7
  - parameters . . . . . 17-7
- services
  - adding . . . . . 17-5, 17-6
  - adding dialogs . . . . . 17-2
  - attaching a dialog . . . . . 17-8
  - character mapping . . . . . A-51

- services (continued)
  - connecting . . . . . 17-5
  - defined . . . . . 17-1
  - device type . . . . . 11-3
  - installing . . . . . 17-5
  - network . . . . . 17-6
  - restricting access . . . . . 17-8
- session file, DOS PC . . . . . 11-40
- Session tab, for keyboards . . . . . 7-18
- setting
  - a purge interval . . . . . G-28
  - idle time-out . . . . . 6-19
  - login time-out . . . . . 6-20
  - traits for groups, read group . . . . . 6-23
  - user passwords . . . . . 18-3
  - write group traits for queues . . . . . 6-24
- setting up forms . . . . . 8-1 to 8-33
- setup options, combining . . . . . 12-6
- SGI IRIX system
  - adding network services . . . . . 17-8
  - shutting down . . . . . 3-12
  - startup . . . . . 3-6
  - stopping commands . . . . . 2-4, 2-13
- shell . . . . . 22-12
- showtimingbar . . . . . F-12
- shutdown console command . . . . . A-35
- shutting down the system . . . . . 3-9 to 3-12
- SILENT constraint level . . . . . 13-34
- single system profile parameter . . . . . 11-28
- site files
  - backing up . . . . . 20-25
  - changing (caution) . . . . . 11-8
- /site/config . . . . . 10-4, B-6
- defined . . . . . 11-7
- /site/dict . . . . . 10-4
- /site/dict/dial . . . . . C-39
- /site/dict/keymacros . . . . . C-40
- /site/dict/messages . . . . . C-9
- /site/dict/printmsgs . . . . . C-43
- /site/dict/queues . . . . . 14-70, C-27
  - for FTS . . . . . 14-70
  - /site/dict/shift . . . . . C-44
  - /site/dict/vtmap . . . . . C-46
  - /site/dict/words . . . . . 14-70, C-29
    - for FTS . . . . . 14-70
  - /site/etc/rc . . . . . 10-4
  - /site/mcs . . . . . 10-4
  - /site/printers . . . . . 10-4
  - /site/printers directory . . . . . 12-2
  - /site/printers/ti830 system file . . . . . B-8
  - /site/system (profile file) . . . . . 11-17
  - /site/system (system file) . . . . . 10-4, B-8
  - /site/wires . . . . . 10-4
  - /site/wires/anpa7 (system file) . . . . . B-9
- sitedump command . . . . . 20-25, A-35
- siterestore console command . . . . . A-36
- skip flags . . . . . 5-39, 20-4, 20-7
- so, system operator prompt . . . . . 3-2
- softdump command . . . . . 20-23, A-36
- softrestore console command . . . . . A-36
- sort . . . . . 5-29
- sort field . . . . . 5-28
- sortfield database trait . . . . . G-25
- sorting stories . . . . . A-14
- source command . . . . . 14-39, 14-47, 14-55
- source job list command . . . . . A-47
- space
  - blocks . . . . . 19-3
  - for font and forms . . . . . 12-7
  - free list . . . . . 19-3
  - hogs command . . . . . 19-5
  - increasing in the database . . . . . 19-7
  - reclaiming . . . . . 19-6
  - usage . . . . . 19-5
- spacing, printer profile . . . . . 12-15
- special characters, in a printer
  - profile . . . . . 12-19 to 12-23
- specifying members of a group . . . . . 6-6
- specifying users for queue notification . . . . . 6-25
- speed parameter . . . . . 11-43
- SPEED parameter, list c command . . . . . 11-7
- SRPl0-LIsUGQsXWFi, explained . . . . . G-18

- standard dictionaries . . . . . C-1
- standard mailbox . . . . . 14-17
- standard programmable keys . . . . . 7-23
- start keyword . . . . . 13-24
- starting
- the console . . . . . 2-14
  - the system . . . . . 3-5 to 3-9
- startup after power failure . . . . . A-7
- startup and shutdown . . . . . 3-5 to 3-12
- startup command . . . . . A-36
- state keys, defined . . . . . 7-9
- status console command . . . . . 11-20
- Status field . . . . . 14-90
- STATUS form field . . . . . 8-22
- status of passwords . . . . . 18-3
- STILL-ID form field . . . . . 8-22
- STILL-PRESET form field . . . . . 8-22
- stop all command . . . . . 5-11
- stop console command . . . . . A-38
- stop dialog command . . . . . A-54
- stopbits keyword . . . . . 13-25
- storage units . . . . . 19-3
- stories
- busy . . . . . 5-52
  - cannot read or write . . . . . 22-5
  - changing database traits . . . . . 5-3
  - default width of scripted story . . . . . 11-28
  - default width of unscripted story . . . . . 11-28
  - embed HTML codes . . . . . 15-14
  - error checking . . . . . A-10
  - getting info . . . . . 5-15
  - how the system copies . . . . . 19-4
  - pointers . . . . . 19-4
  - purging . . . . . 19-2
  - removes old versions . . . . . A-11
  - retrieve backup from tape . . . . . A-12
  - sort . . . . . A-14
  - storing copies . . . . . 19-4
  - transmitting . . . . . 14-89
  - unbusy . . . . . 22-6
  - unlocking . . . . . 5-51
  - when last modified . . . . . 5-49
  - who last modified . . . . . 5-49
  - who moved, duplicated, or killed . . . . . 5-17
- story components . . . . . 15-5
- story entity . . . . . 15-5
- defined . . . . . 15-5
  - formats . . . . . 15-6
  - reference names, rules . . . . . 15-7
- story forms . . . . . 5-28
- compatibility errors . . . . . 5-30
- story forms, HTML . . . . . 15-5 to 15-23
- contents . . . . . 15-20
  - sample . . . . . 15-18
- story notes . . . . . 13-35
- story order, changing . . . . . 14-92
- style story
- changing profile options . . . . . 12-25
  - creating . . . . . 12-24
  - fonts . . . . . 12-4
  - maximum number of styles . . . . . 12-24
  - naming . . . . . 12-24
  - selecting fonts . . . . . 12-26
  - selecting forms . . . . . 12-26
- styles for VT, with local printing . . . . . 12-30
- styles, with local printing . . . . . 12-38
- su prompt... *See superuser prompt*
- subdirectories, apply changes to all . . . . . 5-24
- superuser
- console command . . . . . A-38
  - logging in . . . . . 3-3
  - password . . . . . 3-3 to 3-5
  - prompt . . . . . 3-2
- suppressing a search . . . . . 13-42
- switching keyboards . . . . . 7-26
- symbols and conventions
- console . . . . . xxx
  - in this manual . . . . . xxviii
  - keyboard . . . . . xxix
- synctoserver . . . . . F-16

- system
  - connecting . . . . . 14-88
  - data . . . . . 19-1
  - defined . . . . . 11-16
  - list all computers . . . . . A-33
  - name . . . . . 11-22
  - net parameter . . . . . 11-25
  - security . . . . . 18-1 to 18-12
  - single computer . . . . . 11-28
  - triple . . . . . 11-13
- system access
  - group security . . . . . 6-20 to 6-28, 18-12
- system administrator, tasks . . . . . 1-8 to 1-12
- system characters
  - avoiding use . . . . . 13-27
  - decimal values . . . . . 13-28
- SYSTEM directory
  - hiding . . . . . 6-28, G-36
  - restricting . . . . . 6-28, G-36
- system failure, recovering . . . . . 21-9
- system operator . . . . . 3-2
  - logging in . . . . . 3-2
  - password . . . . . 3-3 to 3-5
  - prompt . . . . . 3-2
- system printing . . . . . 12-2 to 12-30
  - managing printers . . . . . 12-47 to 12-49
- system profile
  - airmargin . . . . . 11-21
  - airshift . . . . . 11-21
  - checking current settings . . . . . 11-20
  - default values . . . . . 11-18
  - highwater . . . . . 11-22
  - id . . . . . 11-22
  - incorporating changes . . . . . 11-20
  - introduction . . . . . 11-17
  - lastlogin . . . . . 11-22
  - load . . . . . 11-22
  - localtimeout . . . . . 11-23
  - logintimeout . . . . . 11-23
  - lowwater . . . . . 11-23
  - master . . . . . 11-24
  - maxhits . . . . . 11-24
  - min\_passwd\_length . . . . . 11-24
  - modifying . . . . . 11-19
  - msgserver . . . . . 11-25
  - name . . . . . 11-25
  - net . . . . . 11-25
  - netlogintimeout . . . . . 11-25
  - outtime . . . . . 11-26
  - parameter list . . . . . 11-21
  - pausetimeout . . . . . 11-26
  - purgelimit . . . . . 11-26
  - readrate . . . . . 11-27
  - remotetimeout . . . . . 11-27
  - scripthmax . . . . . 11-27
  - scriptrhmax . . . . . 11-28
  - security . . . . . 11-28
  - security parameter . . . . . 6-18
  - single . . . . . 11-28
  - textmax . . . . . 11-28
  - time values . . . . . 11-29
  - timechar . . . . . 11-29
  - timer . . . . . 11-29
- system profile files . . . . . 11-17
- SYSTEM.ACCOUNT queue . . . . . 8-27
- SYSTEM.CLIENT.DOS queue . . . . . 11-39
- SYSTEM.CLIENT.WINDOWS . . . . . 11-3, B-11
- SYSTEM.CONFIGURE.301-ACTION . . . . . B-12
- SYSTEM.DIALOG directory . . . . . 17-3
- SYSTEM.DISTRIBUTION . . . . . 14-39
- SYSTEM.FORMS . . . . . 8-3
- SYSTEM.GROUPS . . . . . 6-3, 14-17
  - group checker . . . . . 6-7
  - user security . . . . . 4-35
- SYSTEM.KEYBOARDS . . . . . 7-3, 14-17
- SYSTEM.MAIL.ERROR . . . . . 5-8, 5-9
- SYSTEM.MAIL.OUT . . . . . 5-7
- SYSTEM.MAP . . . . . B-13
- SYSTEM.PRINTERS directory . . . . . 11-42
- SYSTEM.RESOURCE . . . . . 9-5, 9-20, B-15
- SYSTEM.SEARCHTAPE queue . . . . . 20-14
- SYSTEM.SEEK . . . . . 14-62

- SYSTEM.STYLES . . . . . 12-24  
 SYSTEM.TITLE-ENTRY. . . . . 9-5  
 SYSTEM.WIRES.DISTRIBUTION . 13-31, 14-16,  
     B-17  
 SYSTEM.WIRES.KEYWORDS. . . . . 13-40, 14-17,  
     B-18  
 SYSTEM.WIRES.KEYWORDS-AP. . . . . B-19  
 SYSTEM.WIRES.KEYWORDS-AP2. . . . . B-19
- T**
- table formatting, heol command . . . . . A-53  
 tape operations . . . . . 20-2  
 tapes  
   backup . . . . . A-8  
   listing contents . . . . . 20-6, 20-11  
   loading . . . . . 20-2  
   rotating dumps. . . . . 20-2  
   write-protected. . . . . 20-9  
 TAPE-TIME form field. . . . . 8-22  
 tasks  
   creating a job list. . . . . 14-23, 14-38, 14-55  
   for the action server . . . . . 14-23, 14-38, 14-55  
   mailbox. . . . . 14-13  
   multiple . . . . . 14-23  
   timed interval tasks . . . . . 14-18  
 T-connector . . . . . D-3  
 teleprompter  
   airshift . . . . . 11-21  
   screen width parameters . . . . . 11-28  
 telex, dictionary . . . . . C-37  
 text communication . . . . . A-32  
 test configure. . . . . 11-9  
 testing the site configuration file . . . . . 11-9  
 text accents and diacritical marks . . . . . 13-28  
 textmax system profile parameter . . . . . 11-28  
 time and date  
   elements, format strings. . . . . 15-15  
   fields, format strings. . . . . 15-17  
 time values, system profile . . . . . 11-29  
 timechar system profile parameter . . . . . 11-29  
 timed interval tasks  
   defining . . . . . 14-18  
   for Tx links. . . . . 14-96  
   specifying intervals . . . . . 14-19  
 time-out parameters . . . . . 6-19  
 timer  
   dialog command . . . . . A-54  
   setting . . . . . A-54  
 timer system profile parameter . . . . . 11-29  
 timing form . . . . . 8-29  
 TITLE form field . . . . . 8-23  
 title keyword. . . . . 13-25  
 TK-50 tapes . . . . . 20-2  
 tli (temp) keyword . . . . . 13-26  
 top console control command . . . . . 2-23  
 top host section . . . . . 11-14  
 TOTAL-TIME form field . . . . . 8-23  
 tracking database space. . . . . 19-5  
 tracking user login activity . . . . . 18-8  
 transfer queue . . . . . 11-48  
 transferring group assignments . . . . . 6-27  
 translations for dictionaries. . . . . C-3, E-8  
 transmit link... *See Tx link*  
 TRANSMIT option . . . . . 13-34  
 triple systems . . . . . 11-13  
 Tx link  
   adding . . . . . 14-93  
   assigning distribution codes . . . . . 14-34  
   automatic story transmission. 14-93 to 14-100  
   avoiding duplicate stories . . . . . 14-90  
   blacklist . . . . . 14-96  
   changing story order . . . . . 14-92  
   deleting stories after sending. . . . . 14-96  
   introduction. . . . . 14-88  
   job list  
     changing queue order. . . . . 14-96  
     commands . . . . . 14-97  
     queue . . . . . 14-95  
     sample story. . . . . 14-97  
   mailbox tasks . . . . . 14-95

- Tx link (continued)
- network . . . . . 14-88, 14-93 to 14-100
  - network job list . . . . . 14-95
  - queue order . . . . . 14-92
  - relist line . . . . . 14-98
  - security . . . . . 14-96
  - serial . . . . . 14-88
  - validation . . . . . 14-26
- Tx/net
- sending HTML . . . . . 15-2
  - web publishing . . . . . 15-2
- type a string . . . . . A-55
- type dialog command . . . . . A-55
- type, printer profile . . . . . 12-13
- types of devices . . . . . 11-3
- U**
- umanager account, creating . . . . . 4-35
- unbusy command . . . . . 5-52
- unbusyng
- a story . . . . . 22-6
  - Dead queue . . . . . 22-7
  - edit lock . . . . . 22-6
  - order lock . . . . . 22-6
- UNIX commands available in Avstar NRCS . . . . . A-4
- UNIX editor
- defined . . . . . 10-4
  - quitting . . . . . 10-14
  - saving changes . . . . . 10-14
- unlocking a story . . . . . 5-51
- unlocking queues . . . . . 6-21
- up console control command . . . . . 2-23
- update queue . . . . . 14-90
- updating intermediate queues . . . . . 14-91
- uppercase in passwords . . . . . 3-3
- URGENT constraint level . . . . . 13-34
- usage restriction . . . . . 6-20 to 6-28
- user accounts
- adding . . . . . 4-22 to 4-29
  - modifying . . . . . 4-3
  - removing . . . . . 4-33 to 4-34
  - searching . . . . . 4-29 to 4-33
- user activity, tracking . . . . . 18-7 to 18-11
- user manager account, creating . . . . . 4-35
- user passwords, setting and changing . . . . . 18-3
- user preferences, changing . . . . . 4-10
- user traits . . . . . G-3
- blacklist . . . . . 14-96
  - create or kill queues . . . . . 4-8
  - custom toolbars & colors . . . . . 4-8
  - edit mode . . . . . 4-6
  - home queue . . . . . 4-6
  - list u-v command . . . . . G-2
  - mail queue . . . . . 4-7
  - modifying from a workstation . . . . . 4-3 to 4-4
  - password . . . . . 4-8
  - password restrictions . . . . . 18-7
  - read rates . . . . . 4-7
  - reorder stories . . . . . 4-8
  - summary of . . . . . G-11
  - using with database traits . . . . . 5-3
- users
- cannot read or write stories . . . . . 22-5
  - group permissions . . . . . 6-21
  - last login . . . . . 11-22
  - last password change . . . . . 18-6
  - listing log ins . . . . . 18-9
  - login timeout . . . . . 11-23
  - unable to log in . . . . . 22-2
- users without passwords option . . . . . 18-5
- utility messages dictionary . . . . . C-9
- utility programs . . . . . 11-3
- utility programs (servers) defined . . . . . 14-2
- utraits console command . . . . . A-39
- changing a user's password . . . . . 18-5
- V**
- valid block . . . . . 19-3

- validate job list command . . . . . 14-30, A-47
  - validation
    - (action) server messages. . . . . C-13
    - distribution codes. . . . . 14-35
    - ignore yes command . . . . . 14-30
    - introduction . . . . . 14-26
    - job list . . . . . 14-29
    - quiet no command . . . . . 14-30
    - uses. . . . . 14-26
    - validate command . . . . . 14-30
    - with servers or links. . . . . 14-27
  - validation story . . . . . 14-28
    - + and - characters . . . . . 14-29
    - ignore command . . . . . 14-28
    - introduction . . . . . 14-28
  - VAR-N form field . . . . . 8-23
  - verifying a backup tape . . . . . 20-5
  - version console command . . . . . A-40
  - video attribute
    - changing order of appearance . . . . . C-26
    - deleting . . . . . C-26
    - names dictionary . . . . . C-26
  - video dictionary. . . . . C-26
  - video mode
    - mapping fonts . . . . . 12-26
    - standard names . . . . . 12-27
  - VIDEO-ID form field . . . . . 8-23
  - view command, console configuration file . 2-17
  - view console control command . . . . . 2-24
  - VT200 terminal
    - extended macro keys . . . . . 7-24
    - insert char function . . . . . 7-24
  - vtcompatibility . . . . . F-18
  - vtmap. . . . . C-46
- W**
- W\_LOGTYPES value . . . . . 18-10
  - wait dialog command . . . . . A-55
  - Web access. . . . . 16-1
    - installing . . . . . 16-2
  - Web publishing . . . . . 15-1 to 15-23
  - Web story directives, defined . . . . . 15-14
  - wholockedit command . . . . . A-40
  - wildcards
    - destination queue. . . . . 14-33
    - distribution codes . . . . . 14-32
  - wire distribution story . . . . . 13-31 to 13-39
    - adding a distribution line . . . . . 13-35
    - adding keyword search . . . . . 14-59
    - ALWAYS option . . . . . 13-35
    - destination queue. . . . . 13-33
    - disabling priority notification . . . . . 13-34
    - distributing to multiple queues . . . . . 13-35
    - distribution codes . . . . . 14-35
    - distribution line . . . . . 13-31
    - distribution name. . . . . 13-32
    - editing . . . . . 13-8
    - hidden categories. . . . . 13-35
    - instructions . . . . . 14-35
    - mailboxes. . . . . 13-38
    - maximum distribution lines. . . . . 13-38
    - notification priority . . . . . 13-33
    - parallel wire server . . . . . 14-48
    - purge intervals. . . . . 13-38
    - TRANSMIT option. . . . . 13-34
    - unknown wires . . . . . 13-37
  - wire ports/problems. . . . . 22-7
  - wire profile
    - 8-bit characters. . . . . 13-28
    - copying a standard profile. . . . . 13-5
    - creating . . . . . 13-5, 13-11 to 13-31
    - editing . . . . . 13-3, 13-6
    - keywords. . . . . 13-13 to 13-26
      - answer . . . . . 13-13
      - bell . . . . . 13-13
      - bits . . . . . 13-13
      - category . . . . . 13-15
      - distribution. . . . . 13-15
      - end . . . . . 13-16
      - eol. . . . . 13-16



- wire profile: keywords (continued)
    - eop . . . . . 13-17
    - eos . . . . . 13-17
    - flags . . . . . 13-18
    - flash . . . . . 13-19
    - form . . . . . 13-20
    - handshake . . . . . 13-20
    - idle . . . . . 13-20
    - map . . . . . 13-21
    - pagenumber . . . . . 13-22
    - paragraph . . . . . 13-22
    - parity . . . . . 13-22
    - presenter . . . . . 13-23
    - priority . . . . . 13-23
    - reverse . . . . . 13-24
    - start . . . . . 13-24
    - stopbits . . . . . 13-25
    - title . . . . . 13-25
    - tli (temp) . . . . . 13-26
    - accent . . . . . 13-13
    - moving a profile . . . . . 13-5
    - nonprinting characters . . . . . 13-26
    - options . . . . . 13-13 to 13-26
  - wire program messages . . . . . C-13
  - wire service
    - configuration line . . . . . 13-4
    - device type . . . . . 11-3
    - parameters . . . . . 13-4
  - wire story form . . . . . 8-33
  - wiredump console command . . . . . A-41
  - wires
    - 8-bit characters . . . . . 13-28
    - adding . . . . . 11-47 to 11-48, 13-2 to 13-11
    - adding to the configuration file . . . . . 13-2
    - bits per character . . . . . 13-13
    - configuration . . . . . 13-3
    - devices . . . . . 11-3
    - distribution . . . . . 13-31
    - handshaking . . . . . 13-20
    - idle notification . . . . . 13-20
    - keyword searching . . . . . 13-39 to 13-50
    - mapping characters . . . . . 13-21
    - merging multiple takes . . . . . 13-4
    - modifying a profile . . . . . 13-11
    - moving . . . . . 11-47, 13-2
    - notification levels . . . . . 13-36
    - parallel wire servers . . . . . 14-41
    - parity . . . . . 13-22
    - redundancy . . . . . 14-41
    - search job list . . . . . 13-41
    - suppress searching . . . . . 13-42
  - WIRES.ALL notification priority . . . . . 13-36
  - WIRES.UNKNOWN . . . . . 13-37
  - workstations
    - adding devices . . . . . 11-35
    - automatic logout . . . . . 11-23
    - automatic time out . . . . . 6-19
    - configuration . . . . . 11-36
    - definition . . . . . 11-3
    - disabling automatic logout . . . . . 11-23
    - group access restrictions . . . . . 11-28
    - idle timeout . . . . . 6-19
    - problems . . . . . 21-3, 22-2
    - setting idle timeout . . . . . 6-19
    - setting login timeout . . . . . 6-20
  - write changes to disk (w) . . . . . 10-14
  - write group . . . . . 5-35, 6-24
  - write-protect tab on tapes . . . . . 20-2
  - write-protected tape . . . . . 20-9
  - WRITER form field . . . . . 8-24
- X**
- x console control command . . . . . 2-24
  - XOFF message . . . . . 2-8
  - xon/xoff flow control . . . . . 22-9
- Z**
- zoom console control command . . . . . 2-24



# Reader's Comments

Your comments assist us in improving the quality and usefulness of our publications. We welcome any suggestions for improving this manual, and would appreciate your comments regarding discrepancies or omissions you may have discovered.

Please send your comments to:

**iNEWS  
Technical Publications Dept  
6400 Enterprise Lane  
Madison, WI 53713**

**Publication:** Avstar NRCS v1.3 Operations Manual (2 Volume Set)

**Overall Rating:**     Superior         Good         Average  
                                  Below Average     Poor

Please use back of this page or attach another sheet if more space is needed to complete response.

**Technical or Clerical Errors:**  
**(Please specify volume, chapter, and page where necessary.)**

---

---

---

---

**Suggestions for Improvement:**

---

---

---

---

**Submitted by:** \_\_\_\_\_



---

# Notes

---

# Notes