



Aid® iNEWS®
Administration Guide

Legal Notices

Product specifications are subject to change without notice and do not represent a commitment on the part of Avid Technology, Inc.

The software described in this document is furnished under a license agreement. You can obtain a copy of that license by visiting Avid's Web site at www.avid.com. The terms of that license are also available in the product in the same directory as the software. The software may not be reverse assembled and may be used or copied only in accordance with the terms of the license agreement. It is against the law to copy the software on any medium except as specifically allowed in the license agreement.

Avid products or portions thereof are protected by one or more of the following United States Patents: 4,970,663; 5,267,351; 5,309,528; 5,355,450; 5,396,594; 5,440,348; 5,467,288; 5,513,375; 5,528,310; 5,557,423; 5,577,190; 5,584,006; 5,640,601; 5,644,364; 5,654,737; 5,715,018; 5,719,570; 5,724,605; 5,726,717; 5,729,673; 5,745,637; 5,752,029; 5,754,851; 5,799,150; 5,812,216; 5,828,678; 5,842,014; 5,852,435; 5,999,406; 6,038,573; 6,061,758; 6,141,007; 6,211,869; 6,532,043; 6,546,190; 6,596,031; 6,636,869; 6,747,705; 6,763,523; 6,766,357; 6,813,622; 6,847,373; 7,081,900; RE40,107; 7,403,561; 7,433,519; D352,278; D372,478; D373,778; D392,267; D392,268; D392,269; D395,291; D396,853; D398,912.
Other patents are pending.

This document is protected under copyright law. An authorized licensee of Avid iNEWS Command may reproduce this publication for the licensee's own use in learning how to use the software. This document may not be reproduced or distributed, in whole or in part, for commercial purposes, such as selling copies of this document or providing support or educational services to others. This document is supplied as a guide for Avid iNEWS Command. Reasonable care has been taken in preparing the information it contains. However, this document may contain omissions, technical inaccuracies, or typographical errors. Avid Technology, Inc. does not accept responsibility of any kind for customers' losses due to the use of this document. Product specifications are subject to change without notice.

Copyright © 2014 Avid Technology, Inc. and its licensors. All rights reserved.

The following disclaimer is required by Apple Computer, Inc.:

APPLE COMPUTER, INC. MAKES NO WARRANTIES WHATSOEVER, EITHER EXPRESS OR IMPLIED, REGARDING THIS PRODUCT, INCLUDING WARRANTIES WITH RESPECT TO ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.

The following disclaimer is required by Sam Leffler and Silicon Graphics, Inc. for the use of their TIFF library:

Copyright © 1988–1997 Sam Leffler
Copyright © 1991–1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software [i.e., the TIFF library] and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The following disclaimer is required by the Independent JPEG Group:

This software is based in part on the work of the Independent JPEG Group.

This Software may contain components licensed under the following conditions:

Copyright (c) 1989 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Copyright (C) 1989, 1991 by Jef Poskanzer.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. This software is provided "as is" without express or implied warranty.

Copyright 1995, Trinity College Computing Center. Written by David Chappell.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. This software is provided "as is" without express or implied warranty.

Copyright 1996 Daniel Dardailler.

Permission to use, copy, modify, distribute, and sell this software for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Daniel Dardailler not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Daniel Dardailler makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Modifications Copyright 1999 Matt Koss, under the same license as above.

Copyright (c) 1991 by AT&T.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR AT&T MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

This product includes software developed by the University of California, Berkeley and its contributors.

The following disclaimer is required by Nexidia Inc.:

© 2006 Nexidia. All rights reserved.

Manufactured under license from the Georgia Tech Research Corporation, U.S.A. Patent Pending.

The following disclaimer is required by Paradigm Matrix:

Portions of this software licensed from Paradigm Matrix.

The following disclaimer is required by Ray Sauers Associates, Inc.:

"Install-It" is licensed from Ray Sauers Associates, Inc. End-User is prohibited from taking any action to derive a source code equivalent of "Install-It," including by reverse assembly or reverse compilation, Ray Sauers Associates, Inc. shall in no event be liable for any damages resulting from reseller's failure to perform reseller's obligation; or any damages arising from use or operation of reseller's products or the software; or any other damages, including but not limited to, incidental, direct, indirect, special or consequential Damages including lost profits, or damages resulting from loss of use or inability to use reseller's products or the software for any reason including copyright or patent infringement, or lost data, even if Ray Sauers Associates has been advised, knew or should have known of the possibility of such damages.

The following disclaimer is required by Videomedia, Inc.:

"Videomedia, Inc. makes no warranties whatsoever, either express or implied, regarding this product, including warranties with respect to its merchantability or its fitness for any particular purpose."

"This software contains V-LAN ver. 3.0 Command Protocols which communicate with V-LAN ver. 3.0 products developed by Videomedia, Inc. and V-LAN ver. 3.0 compatible products developed by third parties under license from Videomedia, Inc. Use of this software will allow "frame accurate" editing control of applicable videotape recorder decks, videodisc recorders/players and the like."

The following disclaimer is required by Altura Software, Inc. for the use of its Mac2Win software and Sample Source Code:

©1993–1998 Altura Software, Inc.

The following disclaimer is required by Ultimatte Corporation:

Certain real-time compositing capabilities are provided under a license of such technology from Ultimatte Corporation and are subject to copyright protection.

The following disclaimer is required by 3Prong.com Inc.:

Certain waveform and vector monitoring capabilities are provided under a license from 3Prong.com Inc.

The following disclaimer is required by Interplay Entertainment Corp.:

The "Interplay" name is used with the permission of Interplay Entertainment Corp., which bears no responsibility for Avid products.

This product includes portions of the Alloy Look & Feel software from Incors GmbH.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

© DevelopMentor

This product may include the JCifs library, for which the following notice applies:

JCifs © Copyright 2004, The JCIFS Project, is licensed under LGPL (<http://jcifs.samba.org/>). See the LGPL.txt file in the Third Party Software directory on the installation CD.

Avid Interplay contains components licensed from LavanTech. These components may only be used as part of and in connection with Avid Interplay.

Attn. Government User(s). Restricted Rights Legend

U.S. GOVERNMENT RESTRICTED RIGHTS. This Software and its documentation are "commercial computer software" or "commercial computer software documentation." In the event that such Software or documentation is acquired by or on behalf of a unit or agency of the U.S. Government, all rights with respect to this Software and documentation are subject to the terms of the License Agreement, pursuant to FAR §12.212(a) and/or DFARS §227.7202-1(a), as applicable.

Trademarks

003, 192 Digital I/O, 192XD I/O, 888 I/O, Adrenaline, AirPlay, AirSPACE, AirSPACE HD, AirSpeed, ALEX, Alienbrain, AniMatte, AudioMarket, AudioPages, AudioSuite, AudioVision, AutoSync, Avid, Avid Advanced Response, Avid DNA, Avid DNxcel, Avid DNxHD, AVIDdrive, Avid DS Assist Station, Avid EditStar, Avid Learning Excellerator, Avid Liquid, Avid Liquid Chrome Xe, Avid MEDIArray, Avid Mojo, AvidNet, AvidNetwork, Avid NewStar, Avid Remote Response, AVIDstripe, Avid Unity, Avid Unity ISIS, Avid VideoRAID, Avid Xpress, AVoption, AVX, Beauty Without The Bandwidth, Blacktooth, Boom, Cj24, CamCutter, CaptureManager, ChromaCurve, ChromaWheel, Commandj24, Conectiv, Countdown, DAE, Dazzle, Dazzle Digital Video Creator, Deko, DekoCast, D-Fi, D-fx, DigiDelivery, Digidesign, Digidesign Audio Engine, Digidesign Intelligent Noise Reduction, DigiDrive, DigiLink, DigiMeter, DigiSerial, Digital Nonlinear Accelerator, DigiTranslator, DINR, DNxchange, do more, DVD Complete, D-Verb, Eleven, Equinox, EveryPhase, ExpertRender, Fastbreak, Fast Track, FieldPak, Film Composer, FilmScribe, Flexevent, FluidMotion, FXDeko, G7, G-Rack, HD Core, HD Process, HDPack, Hollywood DV-Bridge, Hybrid, HyperControl, HyperSPACE, HyperSPACE HDCAM, IllusionFX, Image Independence, iNEWS, iNEWS Assign, iNEWS ControlAir, Instantwrite, Instinct, Intelli-sat Broadcasting Recording Manager, Intelli-Sat, InterFX, Interplay, inTONE, Intraframe, iS9, iS18, iS23, iS36, ISIS, IsoSync, KeyRig, KeyStudio, LaunchPad, LeaderPlus, Lightning, ListSync, Lo-Fi, Magic Mask, Make Anything Hollywood, make manage move | media, Marquee, M-Audio, M-Audio Micro, Maxim, Mbox, MCXpress, Media Browse, Media Composer, MediaDock, MediaDock Shuttle, Media Fusion, Media Illusion, MediaLog, Media Reader, Media Recorder, MEDIArray, MediaShare, MediaStream, Media Suite, Meridian, MetaFuze, MetaSync, MicroTrack, Midiman, MissionControl, Mix Rack, MixLab, Moviebox, Moviestar, NaturalMatch, Nearchive, NetReview, NewsCutter, Nitris, NRV-10 interFX, Octane, OMF, OMF Interchange, OMM, OnDVD, Open Media Framework, Open Media Management, Palladium, Pinnacle, Pinnacle DistanTV, Pinnacle Geniebox, Pinnacle HomeMusic, Pinnacle MediaSuite, Pinnacle Mobile Media, Pinnacle PCTV, Pinnacle PCTV HD Ultimate Stick, Pinnacle PCTV Nano Stick, Pinnacle PCTV To Go, Pinnacle Scorefitter, Pinnacle Studio, Pinnacle Studio MovieBoard, Pinnacle Systems, Pinnacle VideoSpin, ProEncode, ProServices, ProSessions, Pro Tools, QuietDrive, Recti-Fi, Reel Tape Delay, Reel Tape Flanger, Reel Tape Saturation, RetroLoop, rS9, rS18, Salesview, Sci-Fi, Scorch, Scorefitter, ScriptSync, SecureProductionEnvironment, Session, Show Center, Sibelius, SIDON, Soft SampleCell, Soft-Clip Limiter, Sound Designer II, SPACE, SPACEShift, SpectraGraph, SpectraMatte, Starplay, SteadyGlide, Streamfactory, Streamgenie, StreamRAID, Strike, Structure, Studiophile, SubCap, Sundance Digital, Sundance, Symphony, SYNC HD, SynchroScience,

SynchroScope, Syntax, Targa, TDM FlexCable, Thunder, Titan, Titansync, TL Aggro, TL AutoPan, TL Drum Rehab, TL Everyphase, TL Fauxlder, TL In Tune, TL MasterMeter, TL Metro, TL Space, TL Utilities, Torq, Torq Xponent, Transfuser, Trigger Finger, Trillium Lane Labs, TruTouch, UnityRAID, Vari-Fi, Velvet, Venom, VideoRAID, Video Slave Driver, VideoSPACE, VideoSpin, Vortex, XDeck, X-Form, Xmon, Xponent, X-Session, and X-Session Pro are either registered trademarks or trademarks of Avid Technology, Inc. in the United States and/or other countries.

Footage

Arri — Courtesy of Arri/Fauer — John Fauer, Inc.
Bell South "Anticipation" — Courtesy of Two Headed Monster — Tucker/Wayne Atlanta/GMS.
Canyonlands — Courtesy of the National Park Service/Department of the Interior.
Eco Challenge British Columbia — Courtesy of Eco Challenge Lifestyles, Inc., All Rights Reserved.
Eco Challenge Morocco — Courtesy of Discovery Communications, Inc.
It's Shuttletime — Courtesy of BCP & Canadian Airlines.
Nestlé Coffee Crisp — Courtesy of MacLaren McCann Canada.
Saturn "Calvin Egg" — Courtesy of Cossette Communications.
"Tigers: Tracking a Legend" — Courtesy of www.wildlifeworlds.com, Carol Amore, Executive Producer.
"The Big Swell" — Courtesy of Swell Pictures, Inc.
Windhorse — Courtesy of Paul Wagner Productions.

Arizona Images — KNTV Production — Courtesy of Granite Broadcasting, Inc., Editor/Producer Bryan Foote.
Canyonlands — Courtesy of the National Park Service/Department of the Interior.
Ice Island — Courtesy of Kurtis Productions, Ltd.
Tornados + Belle Isle footage — Courtesy of KWTV News 9.
WCAU Fire Story — Courtesy of NBC-10, Philadelphia, PA.
Women in Sports – Paragliding — Courtesy of Legendary Entertainment, Inc.

News material provided by WFTV Television Inc.

GOT FOOTAGE?

Editors — Filmmakers — Special Effects Artists — Game Developers — Animators — Educators — Broadcasters — **Content creators of every genre** — Just finished an incredible project and want to share it with the world?

Send us your reels and we may use your footage in our show reel or demo!*

For a copy of our release and Avid's mailing address, go to www.avid.com/footage.

*Note: Avid cannot guarantee the use of materials submitted.

Avid iNEWS v5.1 Administration Guide • 9329-65311-00 Rev B • Created 8/21/14 • This document is distributed by Avid in online (electronic) form only, and is not available for purchase in printed form.

Contents

	Using This Guide	19
	Symbols and Conventions	20
	If You Need Help	20
	...With the Syntax of Commands	21
	Avid Training Services	23
Chapter 1	Introduction	25
	Overview of iNEWS	26
	System Administrator Tasks	27
Chapter 2	Connect Services	30
	Network Services	31
	Dialogs for Connect Services	31
	Building a Dialog	31
	Dialog Commands	32
	Dialog Examples	33
	Adding System Services	33
	Setting up the Service	33
	Console Connect Sessions	37
Chapter 3	Database Security	39
	Establishing Security Procedures	40
	User Passwords	40
	Checking Password Status	41
	Forcing Individual Users to Change Their Passwords	42
	. . . At an iNEWS Workstation	43
	Tracking User Activity	44
	Tracking User Login Activity and Date Created	44
	. . . At an iNEWS Workstation	44
	. . . At the Console	44
	Listing Users Currently Logged in	45
	Recording Logins	45

	Using Group Security to Control System Access	47
Chapter 4	Database Management	48
	Monitoring Free Space	49
	Understanding Database Storage Units	49
	Monitoring the Free List	50
	Understanding How the System Copies Stories	50
	Tracking Database Space over Time	51
	Using the hogs Command to Obtain Information	51
	Using dobserver and dbfree to Obtain Information	52
	Increasing Database Space for Immediate Use	52
	Maintaining the Database	53
	Checking the Database for Errors	53
	Cleaning the Database	55
	Cleaning Your Database Offline	56
	Cleaning Your Database with Online Dbvisit	58
Chapter 5	Avid System Monitoring	62
	Using SNMP Monitoring on the iNEWS Server	63
	Installing and Enabling the SNMP Agent	63
Chapter 6	Backing Up the iNEWS System	65
	Tape Operations	66
	Establishing Policies for Backup Procedures	66
	Backing up the iNEWS Database	67
	The dbdump Command	67
	Backing up the Entire Database to Tape	68
	Tapeless Database Backup	69
	Backing up Individual Queues	75
	Notes on Backing up the Database	77
	Restoring Data to the iNEWS Database	77
	The dbrestore Command	77
	Restoring a First-Level Directory	78
	Listing Tape Contents and Backup Dates	79
	Listing Contents of a Tape	79
	Listing Items Dumped on a Particular Date	80

	Listing the Date of Each Backup	81
	Searching a Tape	82
	The searchtape Command	82
	Searching a Tape for Stories.	82
	Searching a Tape by Word(s)	82
	Searching a Tape by Word and Date Range.	83
	Searching a Tape by Word and Day	84
	Searching a Tape by Word and Month	84
	Specifying a Maximum Number of Stories to Search	84
	Checking for Free Space on a Database	85
	Adding Blocks to the Free List.	85
	Restoring the Database	86
	Disaster Recovery Planning	86
	Disaster Recovery Dbdump	86
	Create Minimal dbdump	87
	Backing up Software	87
	Backing up System Site Files	88
	The sitedump Command	88
Chapter 7	Disconnects	89
	Normal System Status	90
	Disconnects.	90
	Detecting a Disconnect	91
	Types of Disconnect.	93
	Causes of Disconnects.	94
	Disconnect Recovery	94
	Overview	94
	Procedures.	95
	Recovery Worksheet	99
Chapter 8	Troubleshooting	101
	iNEWS Workstation Problems	102
	A User Cannot Log in.	102
	A User Cannot Establish a Session	102
	A User Cannot Access an Item	104

Group Permissions	104
Busy Stories	104
Wire Problems	106
Data Receiver	106
Locked Blocks	106
How to Check Process Status (ps Command)	107
Power Failure	108
Network Failure	108
netstat -i Command	109
Output Errors (TX-ERRs)	109
Input Errors (RX-ERRs)	109
iNEWS and FTS Connectivity (ftsdebug Command)	110
Checking Communication between iNEWS and FTS Servers	111
Community	113
Appendix A Command References	114
Programs Invoked by iNEWS	114
Commands Used by Avid Personnel Only	115
Linux Commands Used in iNEWS	115
Console Server Commands	116
broadcast	116
configure	116
connect	117
ctrails	118
dbclean	118
dbclose	119
dbdev and dbsize	119
dbdump	120
dbfree	121
dblines	122
dboriginal	122
dbpurge (Superuser conditional)	123
dbrestore	124

dbserver	126
dbsort	126
dbtraits	127
dbvisit	128
dictionary	130
diskclear (Superuser only)	131
diskcopy	131
doc	131
ed	132
enter	132
force (Superuser only)	133
grpcheck	133
gtraits (Superuser only)	133
help	134
hogs	134
idiff	135
list	136
list B	139
list C	140
list c	140
list d	140
list g	142
list p	142
list q	145
list s	145
list sq	146
list u	146
logout	148
makemontab	148
makeshift (Super user only)	148
maketab	

(Superuser only)	149
mapcheck	150
msgclean	150
offline	153
online	154
otod	154
reconnect	155
remove	156
rename	
(Superuser only)	156
reorder	156
restart	157
searchtape	157
send	158
sendlong	158
shutdown	158
sitedump	
(Superuser only)	159
siterestore	
(Superuser only)	159
startup	160
status	160
stop	161
su	161
unbusy	161
utraits	
(Super user only)	162
version	163
wholockedit	164
Job List Commands	165
at	165
blockmode	165
bpoll	166
bscan	166

charset	166
distribution	166
dup	166
every	167
extension	167
fast	167
ignore	167
ignore-del	167
local	168
mailto	168
move	168
number	168
on	168
open	169
order	169
passive	169
poll	170
publish	170
put	170
quiet	170
remote	171
remove	171
replace	171
scan	171
send-del	171
sendform	172
sendpassword	172
source	172
validate	172
Dialog Commands	172
capture	173
delay	173
diag	174

echo	174
escape	174
expect	175
heol	175
map	175
mapin	176
mapout	176
message	176
pass	176
pause	177
stop	177
timer	177
type	178
wait	178
Appendix B System Files	179
/etc/hosts	180
/site/config	181
/site/system	193
SYSTEM.CLIENT.WINDOWS	194
SYSTEM.COLORS	195
SYSTEM.CONFIGURE.301-ACTION	196
SYSTEM.GROUPS	197
SYSTEM.INTERPLAY	197
SYSTEM.LISTS	197
SYSTEM.MAP	198
SYSTEM.MOS-MAP	199
SYSTEM.RESOURCE	200
SYSTEM.WIRES.DISTRIBUTION	203
SYSTEM.WIRES.KEYWORDS	204
SYSTEM.WIRES.KEYWORDS-AP	205
SYSTEM.WIRES.KEYWORDS-AP2	206
Appendix C Standard Dictionaries	207
Using Dictionaries to Define Messages and Commands	207

Customizing Dictionaries.	209
Changing Default Dictionary Values.	209
Restoring Dictionary Defaults.	211
Utility Messages Dictionary (/site/dict/messages).	211
DBServer Program Messages	213
Disconnect Program Messages	213
Category and Keyword Check Program Messages	213
Keyboard Check Program Messages.	214
Keyboard Check Program Messages for Macros.	215
Grpcheck Messages.	216
Wire Program Messages	218
Mail Server Messages	218
Map Check Program Messages.	218
Validation (Action) Server.	218
Seek Server Messages	219
Last Login Messages	220
Print Server Messages.	220
dbtraits Messages	220
Save Error (Workstation) Messages	220
Queues Dictionary (/site/dict/queues).	221
Words Dictionary (/site/dict/words)	223
Keyboard Macros Dictionary (/site/dict/keymacros)	226
Case-shifting Dictionary (/site/dict/shift)	228
MCS Dictionary (/site/dict/mcs).	230
Device Types Used by Monitor Servers and Drivers	230
Special Strings Recognized by the Monitor Server	231
Error Messages for the Monitor Server	231
Status Reported in Device Status Field	234
Job List Command Dictionary (/site/dict/joblist)	234
D Messages Dictionary (/site/dict/dmessages)	236
S Messages Dictionary (/site/dict/smessages)	236
Appendix D Environment Variables	239
Registry Editor.	239

Environment Variables (Registry Values).....	240
CCColor.....	241
DestinationOrder.....	243
MailLookup.....	244
MsgMailAlert.....	245
PIColor.....	246
RGB Hexadecimal Color Chart.....	247
ShowTimingBar.....	248
SyncToServer.....	252
VT Compatibility.....	253
DisableCommandLine.....	255
Environmental Variables for Servers.....	256
Appendix E Managing Traits at the Console.....	259
Viewing User Traits from the Console.....	259
Modifying User Traits from the Console.....	260
Changing a User's Password.....	262
Listing Users Who Do Not Have Passwords.....	264
User Traits Console Command Summary.....	268
Managing Database Traits from the Console.....	271
Getting Basic Information.....	271
Getting Detailed Information.....	272
Changing Database Traits from the Console.....	272
Changing a Parent Directory Only.....	273
Database Traits Console Command Summary.....	273
Sortfield.....	281
Changing a Queue's Sort Field.....	282
Starting the Queue Sort Function from the Console.....	282
Purge Interval.....	283
Mailbox Attribute.....	283
The dis Column.....	284
FTSindex Attribute.....	286
Interplay Attribute.....	286
Managing Group Traits at the Console.....	287

	Read Group	287
	Write Group	287
	Editorial Group	288
	Notify Group	288
	Restricting Access Using Read and Write Limitations	288
	Removing Directory or Queue Restrictions	290
Appendix F	The Line Editor, ed	291
	Starting ed.	291
	Specifying Lines to Edit	292
	Searching the File	293
	Searching Tips	294
	Editing Commands	294
	Saving Changes	297
	Quitting ed	297

Using This Guide

Congratulations on your purchase of your Avid iNEWS system. It is a vital part of the Avid news system solution that integrates with other Avid systems and third-party equipment to provide an ideal nonlinear workflow and optimize the news production process.

This guide is part of a two-book set designed to keep pace with current advances in the Avid system's news production capabilities. The set—made up of this book and the *Avid iNEWS Setup and Configuration Guide*—is a comprehensive resource of all administrative information you will need to take advantage of the many options available to you.




This guide will lead you through even the most complex procedures with task-oriented instructions. The information provided here builds on basic news production procedures described in the help system and other user-based guides, while adding a complete explanation of all of the tools and techniques required to manage the newsroom computer system, including useful tips, shortcuts, and custom options.



The documentation describes the software features and hardware related to the iNEWS newsroom computer system, which is extremely customizable. Your system might not contain certain features and/or hardware that are covered in the documentation.

Symbols and Conventions

Avid documentation uses the following symbols and conventions:

Symbol or Convention	Meaning or Action
	A note provides important related information, reminders, recommendations, and strong suggestions.
	A caution means that a specific action you take could cause harm to your computer or cause you to lose data.
	A warning describes an action that could cause you physical harm. Follow the guidelines in this document or on the unit itself when handling electrical equipment.
>	This symbol indicates menu commands (and subcommands) in the order you select them. For example, File > Import means to open the File menu and then select the Import command.
▶	This symbol indicates a single-step procedure. Multiple arrows in a list indicate that you perform one of the actions listed.
(Windows), (Windows only), (Macintosh), or (Macintosh only)	This text indicates that the information applies only to the specified operating system, either Windows or Macintosh OS X.
Bold font	Bold font is primarily used in task instructions to identify user interface items and keyboard sequences.
<i>Italic font</i>	Italic font is used to emphasize certain words and to indicate variables.
Courier Bold font	Courier Bold font identifies text that you type.
Ctrl+key or mouse action	Press and hold the first key while you press the last key or perform the mouse action. For example, Command+Option+C or Ctrl+drag.
(pipe character)	The pipe character is used in some Avid product names, such as Interplay Production. In this document, the pipe is used in product names when they are in headings or at their first use in text.

If You Need Help

If you are having trouble using your Avid product:

1. Retry the action, carefully following the instructions given for that task in this guide. It is especially important to check each step of your workflow.

2. Check the latest information that might have become available after the documentation was published:
 - If the latest information for your Avid product is provided as printed release notes, they ship with your application and are also available online.
 - If the latest information for your Avid product is provided as a ReadMe file, it is supplied on your Avid installation CD or DVD as a PDF document (`README_product.pdf`) and is also available online.

You should always check online for the most up-to-date release notes or ReadMe because the online version is updated whenever new information becomes available. To view these online versions, select ReadMe from the Help menu, or visit the Knowledge Base at www.avid.com/readme.
3. Check the documentation that came with your Avid application or your hardware for maintenance or hardware-related issues.
4. Visit the online Knowledge Base at www.avid.com/onlinesupport. Online services are available 24 hours per day, 7 days per week. Search this online Knowledge Base to find answers, to view error messages, to access troubleshooting tips, to download updates, and to read or join online message-board discussions.

...With the Syntax of Commands

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

To view instructions about using a command:

- ▶ Use the following command: `help <command name>`

For instance, type: `help dbvisit` for an explanation of the `dbvisit` command.

The following data appears:

```
dbvisit -<d or v or i> [-r or -c name] [-s] [-l] [block# ...]
'r' for read only
's' for "slow" to eliminate cache usage
'c' use checkpoint partition (forces "-s")
'i' to just validate isam files
'l' to list link count distribution
```



Lengthy console displays might be edited to emphasize only the most important information in this guide. An ellipsis (...) represents portions of the console display not shown in the text. Also, because of the margin limitations of this guide, command lines might appear wrapped

to multiple lines. This does not necessarily indicate the need to press an Enter key. Unless otherwise indicated, commands should be typed on a single line, allowing the computer to wrap the text whenever the command line stretches beyond the screen margin.

Avid Training Services

Avid makes lifelong learning, career advancement, and personal development easy and convenient. Avid understands that the knowledge you need to differentiate yourself is always changing, and Avid continually updates course content and offers new training delivery methods that accommodate your pressured and competitive work environment.

To learn about Avid's new online learning environment, Avid Learning Excellerator™ (ALEX), visit <http://learn.avid.com>.

For information on courses/schedules, training centers, certifications, courseware, and books, please visit www.avid.com/training or call Avid Sales at 800-949-AVID (800-949-2843).

1 Introduction

The iNEWS newsroom computer system is an integrated digital news production system, which provides journalists, producers, directors, writers, and technical personnel with an array of tools to make their jobs easier.

This chapter contains the following main sections:

- [Overview of iNEWS](#)
- [System Administrator Tasks](#)
 - [Basic Tasks](#)
 - [User Tasks](#)
 - [Database Tasks](#)
 - [Security Tasks](#)
 - [Customizing the System](#)
 - [Storage Maintenance Tasks](#)
 - [Device Tasks](#)
 - [Reviewing Default Settings](#)
 - [Troubleshooting](#)

Overview of iNEWS

An iNEWS newsroom computer system provides:

- News gathering from text sources
- News production, including:
 - Story creation and script editing
 - Association of machine control items to script
 - Show planning and creation
 - Show archiving
 - Contact organization and scheduling
- News to air, including:
 - On-air playback control
 - File exporting
 - Internet publishing

Some primary components of iNEWS include:

- Linux-based computers running the iNEWS Server software. In this guide, these host computers are referred to as the iNEWS Servers, or individually as server A, server B, and so forth.
- A Windows-based computer running the PuTTY Command Sender program. This computer is known as the console.
- Windows- or Vista-based computers running the iNEWS client software. These computers are known as iNEWS Workstations.
- Windows-based computers running the iNEWS Data Receiver software, which is used to ingest wires and other text-based research material
- Other peripherals, such as printers and teleprompters.

Additionally, the iNEWS system is capable of interfacing with a wide variety of production devices. Avid iNEWS Command provides a central point of control for numerous video servers and graphics devices, or MOS protocol may be used to send playlists to MOS-compatible playout controllers.

System Administrator Tasks

The following sections describe common system administrator responsibilities and tasks.

Basic Tasks

Before you can customize or maintain the iNEWS newsroom computer system, you must learn several basic tasks, which include:

- Start up or shut down iNEWS Server software, which includes logging out users and taking the system offline.
- Back up a site file before making file modifications.
- Send system administrator commands from the console to one or more of your system's computers.
- Become a console superuser, capable of performing actions that are only accessible to users with superuser permissions.

User Tasks

A user is anyone who can log in to the database and use iNEWS 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 iNEWS NRCS database by creating a new user account.
- Remove user accounts of former employees to prevent improper access to the iNEWS NRCS database.

Database Tasks

The iNEWS 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 iNEWS system database to meet your organization's expanding needs—including setting up rundowns.

- 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 iNEWS 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 iNEWS 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 the System

Your responsibilities regarding customization include:

- Customize command names, message text, and other items by changing their entries in your system's dictionary files.
- Create templates for the CG Title Entry tool.
- Design and assign custom keyboards for users with a unique set of keyboard macros.

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 fix minor problems before they become serious.
- Backup the entire database or portions of it onto tape, so if necessary, the information can be restored to the database later.
- Make a backup copy of files any time you make 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 iNEWS 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.
- Edit site-specific files, 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.
- Set up printer styles so users can print stories or queues in predetermined formats. For example, a director rundown only showing specific rundown fields.
- Set up servers, which are utility programs automatically performing various actions on the 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.
- (Optional) Write dialogs—lists of instructions—for each connect service to automate the connection process. A connect service is a device that connects a user to a remote computer system.

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.

2 Connect Services

Connect services are utilities you can set up to enable iNEWS users to connect to other computers or information services over the network, such as a remote console for your iNEWS newsroom computer system.

This chapter contains the following main sections:

- [Network Services](#)
- [Adding System Services](#)
- [Console Connect Sessions](#)

Network Services

It is useful to set up a service to let you connect to your iNEWS Servers. For instance, setting up network remote connect services lets you perform many administrative tasks from your iNEWS Workstation rather than from the 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:

1. Create a dialog queue.

Each dialog must be in a separate queue in the SYSTEM.DIALOGS directory. The first step in creating a new dialog is to go to SYSTEM.DIALOGS 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 console. See the *iNEWS Setup and Configuration Guide* for more information about adding new directories, queues or stories.

2. Open a new story in the queue and build the dialog.
3. Type a message that will be displayed on the user's workstation, indicating the connection is being established, using the following format:

```
message text
```

The message dialog command displays on the user's screen whatever text you enter in place of the *text* parameter.

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:

Command	Description
<code>delay nn</code>	Use this command to pause the dialog while the service continues connect and login processes. (<i>nn</i> is the number of seconds you want the dialog to pause before continuing to the next command)
<code>type text</code>	Use the type command to send a text message to the device to which the workstation is being connected.

- To send your login data, such as an account number, type **type text** with text being the account number.
- 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.

- 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.

- 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.
- 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 ["Command References" on page 497](#).

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 example, <cr> and <lf> are used to indicate a “carriage 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:

1. Determine whether 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 Avid 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 the [iNEWS Setup and Configuration Guide](#) for more information on the Site Configuration file and selecting servers via the PuTTYCS application.

3. Add the resource to the configuration file on each server in your system.
 - a. Add the device number of the network resource to the `reslist` line in the server's host definition.
 - b. To keep the network resource available if you bring down one of the system servers, add this line to any appropriate alternate host definitions.



The following steps use `ed` to modify system files. If you are unfamiliar with this line editor, see ["The Line Editor, ed"](#) on page 679.

- c. Use `ed` (line editor) to open the configuration file. The display will look similar to:

```
NRCS-A$ ed /site/config
1356
```

The general format for a network resource configuration line is:

```
resource <device #> <resource name> <device name>
```

Parameter	Description
device #	Identifies resource's device number.
resource name	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 the following example, we call the resource <code>net</code> .
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 net - ; 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>
```

The following table contains information on service line formats:

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 /etc/hosts file. A hyphen in this position will force the service to search each server until it finds one that has a resource it can use.
dialog	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.
resource	Name of the resource you want the service to use.
group	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.
command	Specifies the command you want the service to use to handle communication. To create a service using telnet, type the <code>telnet</code> command here. Ensure you specify the program's 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. Use the `ssh` command in the following way:

```
/usr/bin/ssh <servername> -l <username>
```

The following sample entries in SYSTEM.SERVICE let you establish a network connection to your console from an iNEWS Workstation session.

```
a - console console sysops /usr/bin/ssh NRCS-A -l so
b - console console sysops /usr/bin/ssh NRCS-B -l so
```

The resource assigned to the service must be configured on the computer you want to log in to.

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 containing those instructions. See “[Building a Dialog](#)” on page 31 for more information.

6. (Optional) Test your configuration changes on iNEWS. See the *iNEWS Setup and Configuration Guide* for more information on testing the site configuration file.
7. Reconfigure the system.

This causes your system to note changes and incorporate them into appropriate programs. Do the following:



*See the *iNEWS Setup and Configuration Guide* for more information on selecting servers and the superuser mode.*

- a. Select the console window for the master computer (typically server A).
- b. Enter superuser mode.
- c. Type what appears in bold text:


```
NRCS-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:

```
NRCS-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:

```
NRCS-A# configure -s
```

8. When the prompt returns, type what appears in bold text:


```
NRCS-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:

```
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.

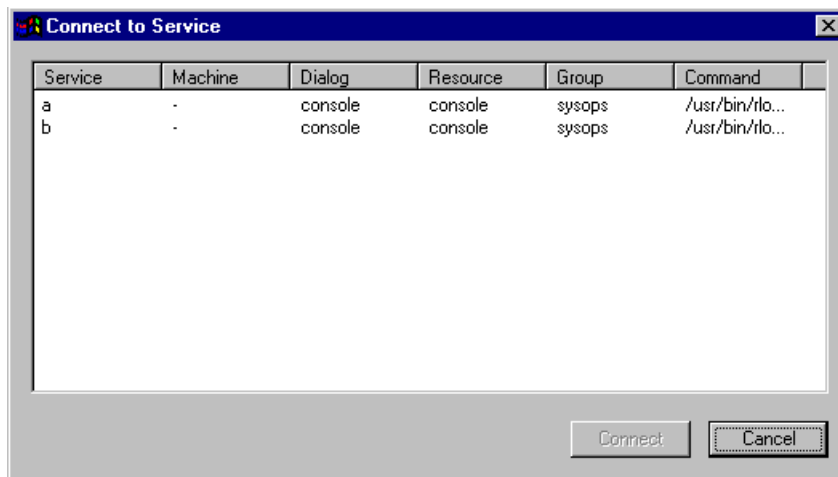
If you have made significant changes, back up your site files with the `sitedump` console command.

Console Connect Sessions

To connect to the console from an iNEWS Workstation—that is, after the network remote connect service and its resources are set up—do the following:

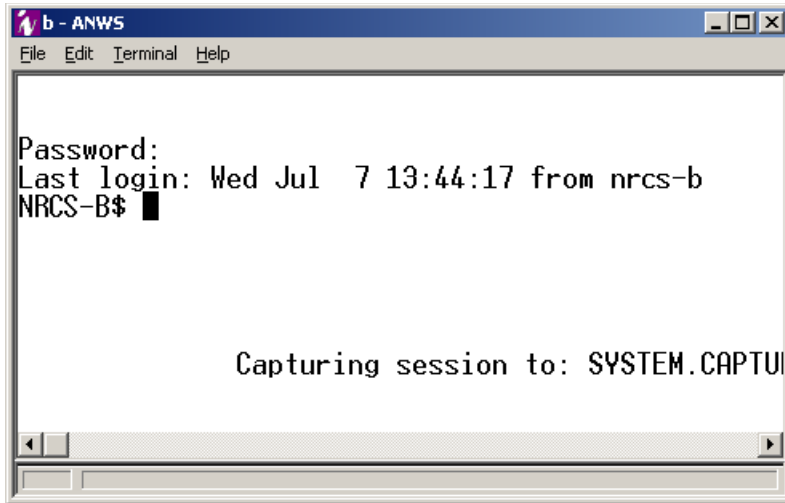
1. Log in to any iNEWS Workstation, with a user account allowed to use the Connect to Service feature.
2. Select Communicate > Connect to Service. The Connect to Service dialog box will appear, offering you a list of services.

As shown in this example, two options are provided to connect to the console—each option corresponding to a different iNEWS Server.



3. Double-click on the service you want.
 - A dialog box will appear with a dialog (as defined in the SYSTEM.DIALOGS) that requests a password.
4. Type in the password and press Enter.
5. Once connected, the dialog box will display the console prompt, similar to what appears on the console when you are logged in as a system operator.

Here is an example of a remote console connection:



6. You can now perform various administrative tasks through the console connect session, as opposed to doing them while physically located at the console.



To stop a console connect session, press Ctrl+D. Do not close the dialog box by selecting File > Close or by clicking on the X button in the upper-right corner.

3 Database Security

This chapter describes how to use various features in the iNEWS newsroom computer system to establish and maintain database security.

This chapter contains the following main sections:

- [Establishing Security Procedures](#)
- [User Passwords](#)
- [Checking Password Status](#)
- [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 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.
- 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 punctuation mark 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.
- If you have dedicated resources, you can 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. This information is kept internally, but having the MODIFY-DEV field makes it visible to users.
- Be familiar with valid user accounts on your system. Restrict access to these account names to protect against unauthorized outside access.

User Passwords

Your iNEWS 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 47](#), lets you control specific areas of your database that each user can access.

For added security, iNEWS uses encryption to pass user credentials to the server.

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 the *iNEWS Setup and Configuration Manual* for more information on setting up new user accounts.

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 iNEWS Workstation.

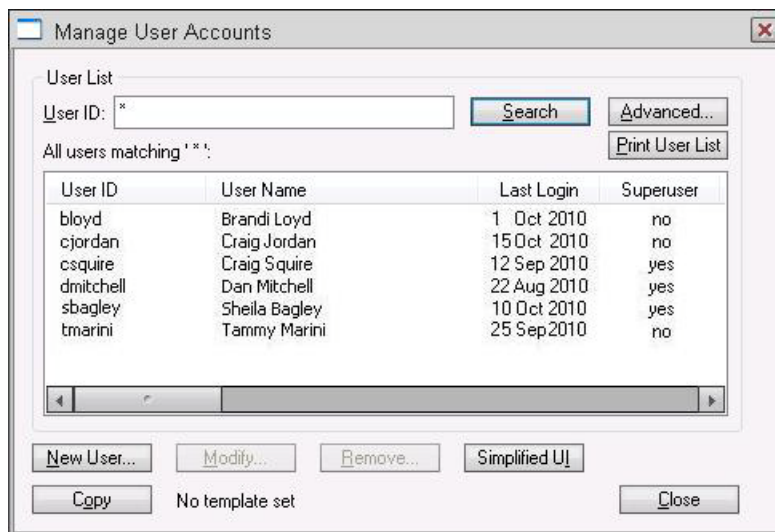


For information on how to check password status from the console, see “Listing Users Who Do Not Have Passwords” on page 652.

To check user password status from an iNEWS Workstation:

1. Log in as a system administrator—that is, use a superuser account.
2. Select Tools > Options > Users.

The Manage User Accounts dialog box appears.



3. Type an asterisk (*) in the User ID field if it does not already appear.

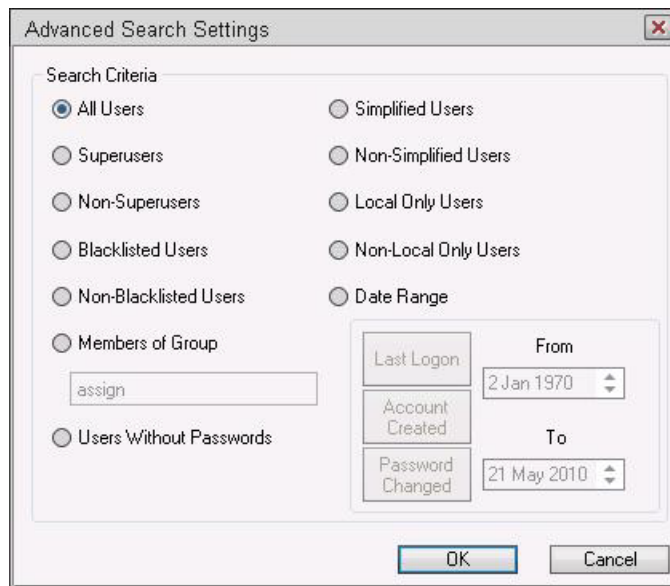
The iNEWS system recognizes this as a wildcard and therefore will search the entire database of users.



The other criteria options available in this dialog box are explained in the iNEWS Setup and Configuration Manual.

4. Click the Advanced button.

The Advanced Search Settings dialog box appears with All Users selected by default.



5. Do one of the following:
 - ▶ To search for all users without passwords, select Users Without Passwords.
 - ▶ 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.
6. Click OK to confirm your advanced search setting or click Cancel to cancel it.
7. 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 the *iNEWS Setup and Configuration Manual* for more information on how to do this from any iNEWS Workstation. For steps on how to change a user's password from the console using the `utraits` console command, see [“Changing a User's Password” on page 650](#).

Forcing Individual Users to Change Their Passwords

Occasionally, you might 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 iNEWS 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 console. For steps on how to force password changes from the console, see [“Changing a User’s Password” on page 650](#).

... At an iNEWS Workstation

To force an individual user to change his or her password:

1. Select Tools > Options > Users.
The Manage User Accounts dialog box appears.
2. Type the user name in the User ID field.
3. Click Search.
4. Select the user name when it appears in the dialog box.
5. Click Modify.

The Modify User Account dialog box appears.

6. Check the Force Change box.
7. 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 iNEWS and when they use it by following the procedures described in this section. These security measures can ensure there is no unauthorized use of your system.

At an iNEWS Workstation, you can determine:

- Last login date of one or more user accounts
- Date user accounts were created
- Users currently logged in

At the 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 iNEWS Workstation.

... At an iNEWS Workstation

To search for a user's last login:

1. Select Tools > Options > Users.
The Manage User Accounts dialog box appears.
2. Click Advanced. The Advanced Search Settings dialog box appears.
3. Click Date Range.
4. Do one of the following:
 - ▶ To search for user accounts with a last login date that matches a specified date range, click Last Login.
 - ▶ To search for user accounts created within a specified date range, click Account Created.
5. Specify a date range in the From and To fields.
6. Click OK.
7. Click Search.

The requested user name information appears.

... At the Console

Another command you can type at the 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 iNEWS Workstation, 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:




1. Press Ctrl+L or select Communicate > Messages > Logged In Users.
A dialog box appears with a list of users currently logged in.
2. Click OK when you are done looking at the list.

To determine whether a specific user is logged in:

1. 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.
4. One of three symbols appears to the left of the To field, depending on what you type in the To text box:

Icon	Description
	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:

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/G
```

Each letter in the W_LOGTYPES value represents a different type of device that can log in on an iNEWS newsroom computer system.

The letters are the same ones that appear in the first column of a list s or list c display for these device types.

2. To track logins by device type, add the appropriate letter to the W_LOGTYPES value.

Use the following letters:

Letter	Description
G	iNEWS Workstation
g	Web Client
B	Web sessions
C	Community sessions
c	COM
A	Interplay (Instinct or NewsCutter)
a	Web Services API
I	Instinct

For instance, to track logins from all devices on your system, change the W_LOGTYPES line to look like this:

```
W_LOGTYPES/GBC
```

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 console:

```
Apr  5 16:25:47 NRCS-A G3494: Attempted login bloyd 172.24.96.247
3.1.1.2/3.1.1.3 RH5
```

This message includes the device type and number, time of login, user name, IP address of PC used, and client version number.

4. A logout message similar to the following is sent to the console when the user logs out:

```
Apr  5 16:23:55 NRCS-A G3495: Logout fiatala 172.24.96.247 {9}
3.1.1.2/3.1.1.3 RH5
```

A failed login—that is, an invalid user name or password—produces a message similar to this:

```
Apr  5 16:25:47 NRCS-A G3494: Attempted login bogus 172.24.96.247
3.1.1.2/3.1.1.3 RH5
```

Regardless of the W_LOGTYPES value, a message is always sent to the 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.

A listing of currently logged in users may be obtained from the console using the list s command. See “list s” on page 529 for more information.

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’s powerful security features let 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 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 iNEWS involve establishing groups and assigning privileges and restrictions to them. The *iNEWS Setup and Configuration Manual* gives you complete information about using the group features in iNEWS to help you maintain system security.

4 Database Management

This chapter provides you with information required to manage the iNEWS database, where all business data, such as stories, is stored. Much of the iNEWS system data is stored in configuration files, which is covered in the *Avid iNEWS Setup and Configuration Guide*.

This chapter contains the following main sections:

- [Monitoring Free Space](#)
- [Tracking Database Space over Time](#)
- [Increasing Database Space for Immediate Use](#)
- [Maintaining the Database](#)

Monitoring Free Space

The iNEWS 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 avoid running out of disk space, iNEWS 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 50](#).

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 reference to the block containing the next part of the story.

A block is either “in use” or “free”.



iNEWS versions 3.x and earlier supported a maximum database size of 16GB. Beginning with iNEWS versions 4.x and later, the maximum database size increased to 128GB. When the database size was expanded the storage block size also increased from 1k blocks to 4k blocks.

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 *lowwater mark*, which represents the least amount of free space available. When the space available drops below the lowwater mark, the system runs `dbserver` to reclaim enough space from the Dead queue to rebuild the free list to the *highwater mark*.

Together, the highwater and lowwater marks determine the free list's normal operational range. If the system cannot get back up to the highwater mark 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 `iNEWS` 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 reference to the original story in the queue that will hold the copy.

When someone opens a copy of a story, the system uses the reference 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 reference.

In most cases, you can treat a reference to an actual story as if it were the story. The only time you need to take references into consideration is when setting purge intervals. When the system references a story to several different queues, each reference takes on the purge interval of the queue. When a reference becomes older than its queue's purge interval, the computer puts the reference, not the actual story, in the Dead queue.

The space used for a story becomes free only when the last reference to the story has been removed from the system.

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	PURGE	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

The USED column contains the number of blocks used in that queue. The SHARED column indicates the number of blocks used in that queue that are also used in other queues. To get the number of blocks used only by that queue, subtract the SHARED count from the USED count. If any queue has a substantially greater number of used blocks than the others, examine that queue more closely.

To obtain a hogs report on the entire database, type:

```
hogs .
```

To send a hogs report to yourself, type:

```
NRCS-A# sh
```

```
# hogs . | mail <your username> &
```



For more examples on how to use the hogs command, see “hogs” on page 518.

Using dbserver and dbfree to Obtain Information

To obtain an accurate idea of how much space is being used by stories in the system:

1. Empty out the Dead queue and reclaim all space used in it. Type:

```
dbserver 3400000
```



You must run dbserver when other programs like dbpurge are not running. The number 3400000 reflects the maximum database size in blocks and ensures the DEAD queue is emptied.

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:

To increase free space for immediate use:

1. Type the command that appears on bold:

```
NRCS-A$ dbfree
```

```
database size 16,386,300 free 1,746,564 = 11%, freelist 1,493
```

The `dbfree` command—used by a system operator or a superuser—displays database size, free list size, and percentage of remaining database available. The freelist number indicates how many database blocks are used to hold the free list. The free list is compressed. This results in anywhere from 250 to 5,000 free-block numbers contained in each free list block. In this example, 1,493 blocks are used to hold the 1,746,564 free block numbers. That is an average of 1,169 free block numbers per free list block.

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 console, type:

```
list d wires
```

Information similar to the following appears:

```
SRPlo-LIisUG-QSXWFiT 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 the *Avid iNEWS Setup and Configuration Guide*.

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 while the system is online. It can point out problems or errors in the structure of the database, find queue corruption that must be fixed before a `dbvisit` is run, and make some repairs. It cannot reclaim unreferenced blocks, adjust link counts, or rebuild the free list; `dbvisit` is required for that.

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, you can have the system send its messages to a log file for you to review the next day.



For more information on the various commands used in this section, see “Command References” on page 497.

To execute the command and have it send output to a log file:

1. Enter superuser mode on the master computer.
2. Type: `dblines . > /tmp/dblines.log`

You can also use the `at` command to schedule the `dblines` command to run at a specified time. For instance:

To execute the `dblines` command so it logs at a specified time:

1. Enter superuser mode on the master computer
2. Get the current time, using the `date` command:

```
NRCS-A# date
Sat Jul 10 11:05:07 EDT 2010
```



You must use 24-hour military time.

3. Set the schedule, such as 5 minutes later, using the `at` command:

```
NRCS-A# at 11:10
```

4. Type the `dblines` command line:

```
/exc/dblines . > /tmp/dblines.log
<Ctrl+D>
```

Pressing `Ctrl+D` after entering the command line marks the end of commands to be executed according to the `at` schedule. Afterwards, the server prompt returns to the original prompt, showing the server name, such as `NRCS-A#`.

After the dblines is complete, you can check the log file for error messages, by using the `grep` command:

```
NRCS-A# grep failed /tmp/dblines.log
```

This searches the log file for the word failed. If there are no errors involving queue corruption, such as `<queue> scan failed`, `<queue> read error`, or `unable to open <queue>`, the prompt will simply be returned. If there are any error messages related to queue corruption, call Avid for assistance.

If you want to review the entire log, you can use the `doc` command to place a copy in a queue you specify. For instance, replacing the previous example, the sequence of commands (shown in bold) may appear similar to the following:



You must use 24-hour military time.

```
NRCS-A# at 11:10
```

```
/exc/dblines . |/exc/doc-ptu people.sysadmin.notes
```

```
<Ctrl+D>
```

```
job 1001340600.a at Sat Jul 10 11:10:00 2010
```

```
NRCS-A#
```

In the example, the system would execute the `dblines` command at 11:10, and then place the output in `PEOPLE.SYSADMIN.NOTES`. As indicated by the pound sign (#) in the prompt, you must be in superuser mode to execute the commands.

Cleaning the Database

Over the course of a month, the iNEWS 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.

You can conduct database cleanup, using the `dbvisit` command, while your system offline or online. The procedures for each are provided in the following sections:

- [Cleaning Your Database Offline](#)
- [Cleaning Your Database with Online Dbvisit](#)

Cleaning Your Database Offline

The database cleanup procedure using the `dbvisit` command requires you to briefly take your system offline, preventing users from logging in.

The day before performing the monthly database cleanup offline, do the following:

1. Select the console window for the master computer (typically server A).
2. Type `dbserver 16000000` to empty the Dead queue.

Starting the day before, allows time for the system to complete this process, which may take several hours, before you run `dbvisit`.



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:



For more details on the procedure for shutting down the system, see the [Avid iNEWS Setup and Configuration Guide](#).

- a. From the PuTTYCS application, select the PuTTY Filter created for sending server commands to all servers, such as iNEWS Consoles.
- b. Take the system offline.
- c. 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`.

- d. Run the stop all console command on all servers to stop all workstations, wires, and other devices from making further changes to the database.

After using the stop all command, wait a few seconds for the prompt to reappear. When it does, proceed to the next step.



If conducting a `dbvisit` offline, ensure system is offline and that you have stopped all network devices. If you do not do this before cleaning your database, you may corrupt the data when you perform the cleanup. For procedures to conduct an online `dbvisit`, see “Cleaning Your Database with Online `Dbvisit`” on page 58.

2. Start the database cleanup by doing the following:
 - a. Select the console window for the master computer (typically server A).
 - b. Enter superuser mode.
 - c. Type: `dbvisit -d`

A message similar to the following appears:

```
09:31:45 opening database
```

```
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 1,000 stories in a queue. After 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.

Add the `-l` flag (the letter L) to include the link count distribution information in the output. If there are link count discrepancies, the system will list them. Such advisory messages about high and low link counts are not errors, however, you will be prompted to correct them.



Running `dbvisit` on all servers instead of just the master computer is an option, performed by typing `dbvisit -dr`. The 'r' indicates read-only mode, and it can run while the read-write `dbvisit` is running on the master server. The read-only `dbvisit` cannot be started until the read-write `dbvisit` reaches the 'traversing directory' phase. After `dbvisit` runs on all computers, the output can be compared to verify the integrity of the mirror.

The `dbvisit` process has four main phases:

- Visiting the “root” blocks
- Traversing the project, directory and queue structures
- Visiting the stories
- Updating the free list and fixing link counts

3. Beginning with iNEWS version 4.x, `dbvisit` was modified so that it would no longer allow you to update the free list if critical errors are found in the database. Instead, if critical errors are found, `dbvisit` will attempt to traverse the directories, reporting all errors it finds, and then completes without offering an opportunity to update the free list. Errors reported during the traversal of stories are not considered critical. Therefore, if the system presents you with the question: Update the free list (y/n)?, do one of the following:
 - ▶ Type **y** to update the free list.
 - ▶ Type **n** to skip the update.



If there are many story errors, you may want to skip the update and contact Avid to assess the problems.

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 that cannot be used until they are put on the free list. Skipping this update 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.

4. Restart your system, by doing the following:
 - a. From the PuTTYCS application, select the PuTTY Filter created for sending server commands to all servers, such as iNEWS Consoles.
 - 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 device cannot be started, a failed message is displayed for that device. Wait for the other devices to start, then use standard procedures to restart failed devices, as covered in the *Avid iNEWS Setup and Configuration Guide*.

Cleaning Your Database with Online Dbvisit

If performing a dbvisit offline prevents usage of your system for too long a period of time, there is a way to run dbvisit while your system is online, letting system administrators do crucial database maintenance without stopping iNEWS users from doing their work.

The strategy used to allow an online dbvisit is to checkpoint the system in a quiet/inactive state, and to maintain the checkpoint state by copying any block that is to change in the database onto the checkpoint partition, namely /dev/rp5backup prior to writing the block onto the database partition - /dev/rp5.

Your system must be shutdown briefly before initiating checkpoint mode. But after checkpoint is turned on, you can bring your system online for iNEWS users while performing database cleanup with the dbvisit command.



Do not conduct an online dbvisit without checkpoint mode initiated. If you do, not do this before cleaning your database, you may corrupt the data when you perform the cleanup.

To perform an online dbvisit:

1. Bring the iNEWS system down and close the database on all servers. This is only for the very brief period of time it takes to ensure checkpoint mode is active.
 - a. From the PuTTYCS application, select the PuTTY Filter created for sending server commands to all servers, such as iNEWS Consoles.
 - b. Type **offline** to take the system offline.
 - c. Type **logout all** to logout all users. You might want to broadcast a message to users prior to this so they can save their work and prepare for the brief downtime.
 - d. Type **shutdown** to stop all devices and close the database.
 - e. The system will present a warning and question: Do you really want to do this? (y/n)? Type **y** to shut down the system.
2. Initiate checkpoint mode.
 - a. (Optional) Type the **status** command to determine which server is the master computer and view the current status of the database.
 For instance:

```
WAVD-A# status
A is OFFLINE and has been CONFIGURED. ID is WAVD.
System is AB. Master is A.
Disk status is OK. The database is CLOSED.
```
 - b. Select the console window for the master computer (typically server A).
 - c. Type **checkpoint on** to reopen the database in checkpoint mode.



Attempting to turn checkpoint on while the database is open will result in the following warning: Database must be CLOSED when activating checkpoint --

3. Bring the iNEWS system online.
 - a. From the PuTTYCS application, select the PuTTY Filter created for sending server commands to all servers, such as iNEWS Consoles.
 - b. Type **startup** to start the system. This will put the system online as well as start up all iNEWS devices.
 - c. (Optional) Type the **status** command to view the current status of the database.

For instance:

```
WAVD-A# status
```

```
A is ONLINE and has been CONFIGURED. ID is WAVD.
```

```
System is AB. Master is A.
```

```
Disk status is OK. The database is OPEN with checkpoint.
```

Users can now log in to iNEWS.

4. Start the database cleanup by doing the following:
 - a. Select the console window for the master computer (typically server A).
 - b. Enter superuser mode.
 - c. Type: **dbvisit -cd**

A message similar to the following appears:

```
09:31:45 2013-06-30 Using checkpoint partition
```

```
09:31:45 traversing roots
```

```
-----message truncated for documentation brevity-----
```

The dbvisit process has four main phases:

- Visiting the “root” blocks
- Traversing the directory, queue, and project structures
- Visiting the stories
- Updating the free list and fixing link counts



More details on the dbvisit command syntax and all of its flags are provided in [Command References appendix](#) of this guide. See “dbvisit” on page 512 for more information.

5. Beginning with iNEWS version 4.x, dbvisit was modified so that it would no longer allow you to update the free list if critical errors are found in the database. Instead, if critical errors are found, dbvisit will attempt to traverse the directories, reporting all errors it finds, and then completes without offering an opportunity to update the free list. Errors reported during the traversal of stories are not considered critical. Therefore, if the system presents you with the question: Update the free list (y/n)?, do one of the following:
 - ▶ Type **y** to update the free list.
 - ▶ Type **n** to skip the update.



If there are many story errors, you may want to skip the update and contact Avid to assess the problems.

6. At the end of the dbvisit process, you might be presented with the question: Do you want to turn off the checkpoint (y/n)?

- ▶ Type **y** to turn checkpoint mode off.

A message similar to the following appears:

```
09:51:17 Checkpoint partition closed
09:51:17 complete - Elapsed time: 20:32
```

7. (Optional) Type the **status** command to view the current status of the database.

For instance:

```
WAVD-A# status
```

```
A is ONLINE and has been CONFIGURED. ID is WAVD.
```

```
System is AB. Master is A.
```

```
Disk status is OK. The database is OPEN.
```

5 Avid System Monitoring

This chapter provides you with information about monitoring iNEWS Server statistics via an SNMP monitoring application such as Avid System Monitor (ASM).

This chapter contains the following main sections:

- [Using SNMP Monitoring on the iNEWS Server](#)
- [Installing and Enabling the SNMP Agent](#)

Using SNMP Monitoring on the iNEWS Server

Avid iNEWS version 4.5 supports SNMP monitoring. SNMP monitoring applications, such as Avid System Monitor, lets you monitor system level environmental information dynamically through an SNMP agent. Avid System Monitor also lets you graph this information over time. For example, the following data can be graphed over time in the Avid System Monitor web page interface:

- Average CPU load
- Number of CPU interrupts per second
- System uptime
- Swap space (disk space reserved for memory when RAM is fully loaded)
- System memory usage
- CPU usage

By default, the SNMP Agent is not enabled.

Installing and Enabling the SNMP Agent

This section describes how to install and enable the SNMP agent on the iNEWS Server. See your Avid representative for information about setting up an Avid System Monitor server and configuring it to monitor the iNEWS Server.



These steps should be performed only by qualified Avid personnel.

To install the SNMP agent on the iNEWS server:

1. Install the net-snmp Red Hat 6 packages. This will create a /etc/snmp directory.
2. Set up the snmpd.conf configuration file. In the /etc/snmp directory, locate the snmpd.conf file. Using a Linux editor, open the configuration file and add the following information near the top of the file.
 - a. Replace the public value in `rocommunity public` with your community name.
 - b. Locate the line with `syslocation` and add your location, such as the name of your city.
 - c. Locate the line with `syscontact` and add your iNEWS administrator email.
3. Save the changes and exit.

To enable the SNMP agent process:

- ▶ In a console to the server, type:


```
# service snmpd start
```

- ▶ (Optional) Set `snmpd` to automatically start at boot by typing:

```
# chkconfig snmpd on
```

To disable the SNMP agent process:

- ▶ In a console to the server, type:

```
# service snmpd stop
```


6

Backing Up the iNEWS System

Proper maintenance of the iNEWS system requires three kinds of backups:

- Database backups—of the iNEWS database
- Software backups—of iNEWS software and the operating system
- System file backups—of files like `/site/config` and `/etc/hosts`

This chapter explains the procedures for the various types of backups necessary to maintain the iNEWS system. It also provides general information about using tapes and tape drives or conducting a tapeless backup for iNEWS on RedHat Enterprise Linux (RHEL).



Previous versions of iNEWS worked on 32-bit or 64-bit RHEL; iNEWS v4.0 and later are only qualified on 64-bit RHEL.

This chapter contains the following main sections:

- [Tape Operations](#)
- [Backing up the iNEWS Database](#)
 - [Backing up the Entire Database to Tape](#)
 - [Tapeless Database Backup](#)
 - [Backing up Individual Queues](#)
- [Restoring Data to the iNEWS Database](#)
- [Disaster Recovery Planning](#)
- [Backing up Software](#)
- [Backing up System Site Files](#)

Tape Operations

When making backup tapes, pay particular 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 may 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 found quickly, 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 might 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 might 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, Avid recommends separate backups be made on each iNEWS 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 79](#) and [“Searching a Tape” on page 82](#).

Establishing Policies for Backup Procedures

Since your software and system files change infrequently, you do not need to back them up very often. Your database, however, changes hundreds or even thousands of times every day, so they need more frequent backups

Avid recommends your station establish and follow a local backup policy for each type of data in your iNEWS system.

The following list has some general backup guidelines:

- Since the database is mirrored across multiple servers you have built-in hardware redundancy. Backups also provide some protection against the 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 it’s own directory.

- If you archive old shows on your system, you might want to produce dbdump tapes that cover old years. For instance, ARCHIVE.2008 through ARCHIVE.2009, 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 use off-site storage services for backup tapes.

Backing up the iNEWS Database

Database backups provide insurance against system calamities. Also, to free up space in the database, you can back up stories to a tape—or to a file, which can be transferred to another computer for storage—and then remove those stories from the database.

Avid recommends making frequent backups of important material and the entire database. To back up database items, you will need to 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:

```
NRCS-A# dbdump c
```

The **c** stands for create. When you use this command, it overwrites any information currently on the tape and 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. 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 the Skip Backup database trait—see the iNEWS Newsroom Computer System Setup and Configuration Manual.

For instance, to dump all database items contained in the SCRIPTS.2010 directory to tape, you could type:

```
dbdump c -n script.2010
```

If you already have dbdump material on tape and you want to append another dbdump to the end of it, use the following command:

```
NRCS-A# dbdump ad
```

An append does not overwrite dbdump information currently on the tape. See [“Backing up Individual Queues” on page 75](#) for more information.



To minimize the impact of any potential problems, perform dbdump at less critical usage periods.

Backing up the Entire Database to Tape

This section provides the procedure for backing up the iNEWS database to a tape. For information on tapeless database backups, see [“Tapeless Database Backup” on page 69](#).

To back up the entire database to tape:

1. Insert a tape into a server’s tape drive.
2. Select the server that has the backup tape. For information on selecting a server, see the *iNEWS Newsroom Computer System Setup and Configuration Manual*.
3. Type **dbdump c**.

A verification request similar to the following appears:

```
Do you really want to create a new archive? (n/y)
```



When backing up the database onto a tape, anything on that tape is overwritten by the new copy of the database.

4. 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. You can then verify the backup. Otherwise, remove the tape from the drive. Set the tape’s write protect switch to the on position, return it to its case, and write the date on the case label.

5. (Optional) Type `dbrestore tdv` at the 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.

A truncated example of 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.

6. When the list is complete, remove the tape.
7. Set the tape's write protect switch to the on position, and write the date on the case label.

Tapeless Database Backup

Instead of backing up the iNEWS database to tapes for potential restoration, you can dump the information to a file and then transfer (FTP) the backed up data to another computer for storage. The more compact way to do this is to use the `dbdump C` command piped through `gzip` and directed to create a file in the `/tmp/` directory. For instance, type:

```
NRCS-A$: dbdump C -f+ |gzip > /tmp/061709.gz
```

To dump information to a file without the gzip option:

1. On an iNEWS Server, use the dbdump command in the following format:

```
dbdump Cv -f <path/file name>
```

For instance, type: NRCS-A\$ **dbdump Cv -f /tmp/dbdump.061709**

The C option creates a new dump file, and the v option switches on the verbose mode, which gives you an idea of progress. After you enter the command, a message similar to the following is displayed:

```
Starting Dump(1) Block(0), dumped on 2009-06-17 16:33:37
```

```
Dumping ISAM files
```

```
Dumping user
```

```
Dumping message
```

```
Dumping group
```

```
Dumping system-blobs
```

```
Dumping community
```

```
Type  Stories      Name
```

```
Dir
```

```
Dir          SHOW
```

```
Dir          SHOW.TRAINING
```

```
Que          SHOW.TRAINING.RUNDOWN
```

```
Que          SHOW.TRAINING.CG
```

```
Que          SHOW.TRAINING.VIDEO
```

```
Que          44      SHOW.TRAINING.MASTER
```

```

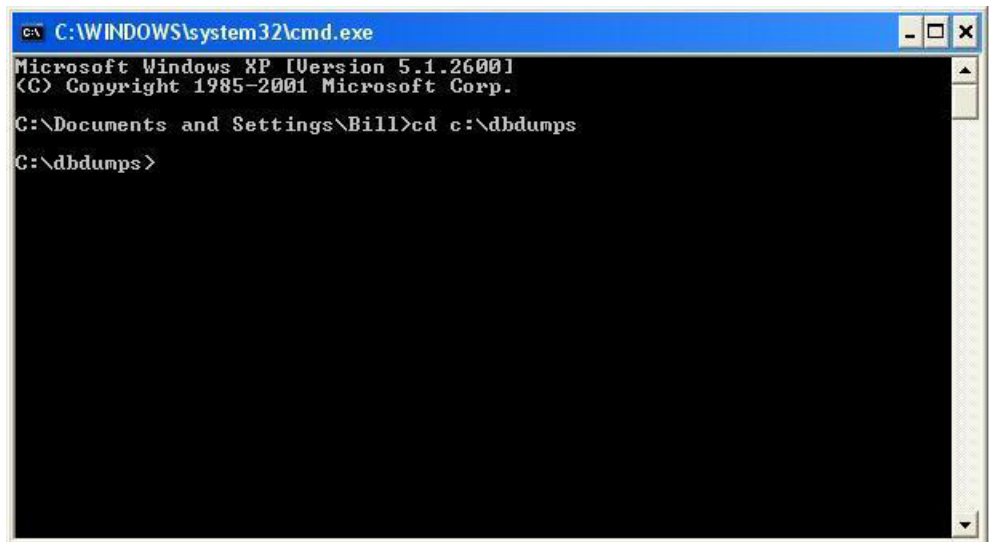
...
Que          1      SHOW.MOSGWE
Que        31840    SHOW.VID-ID
Dir          SHOW.SCRIPTWRITING
Que          SHOW.SCRIPTWRITING.ONE
Que          SHOW.SCRIPTWRITING.TWO
Que          SHOW.SCRIPTWRITING.THREE
Que          5      SHOW.NEWSTARTEST
Que          ARCHIVE.SCRIPTS.2009

310 directories scanned
1684 queues dumped
3109046 stories dumped

Ending Dump(1) Block(136), dumped on 2009-06-17 16:33:37

```

2. On a Windows-based PC with at least 16 GB free for storage, start a command prompt session, by selecting Start > Run and then entering **CMD**.
3. Navigate to the directory on the PC where you want to store the backup file by typing **cd** and the directory path. In this example, the directory is `C:\dbdumps`.



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Bill>cd c:\dbdumps
C:\dbdumps>

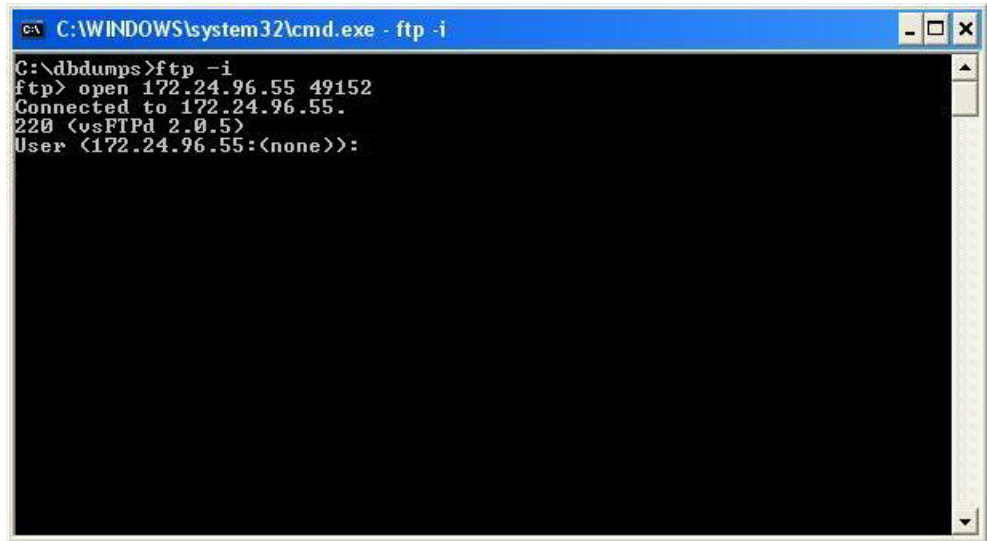
```

4. At the cmd prompt, type **ftp -i** and press Enter to start the FTP session.

- Then type the open command using the following format:

```
open <resolvable name or IP of iNEWS Server> 49152
```

The numerical value at the end of the command line is the number for the port, which should be 49152.



```
C:\WINDOWS\system32\cmd.exe - ftp -i
C:\dbdumps>ftp -i
ftp> open 172.24.96.55 49152
Connected to 172.24.96.55.
220 (vsFTPd 2.0.5)
User (172.24.96.55:(none)):
```

- Type **so** and press Enter. Then enter the system operator password when prompted. A message will appear to indicate whether the login was successful.



```
C:\WINDOWS\system32\cmd.exe - ftp -i
C:\dbdumps>ftp -i
ftp> open 172.24.96.55 49152
Connected to 172.24.96.55.
220 (vsFTPd 2.0.5)
User (172.24.96.55:(none)): so
331 Please specify the password.
Password:
230 Login successful.
ftp>
```


7. Type `cd /tmp` to change to the /tmp directory.
8. Type `binary` to change to binary mode.

A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe - ftp -i". The window shows the following text:

```
C:\dbdumps>ftp -i
ftp> open 172.24.96.55 49152
Connected to 172.24.96.55.
220 (vsFTPd 2.0.5)
User (172.24.96.55:(none)): so
331 Please specify the password.
Password:
230 Login successful.
ftp> cd /tmp
250 Directory successfully changed.
ftp> binary
200 Switching to Binary mode.
ftp>
```

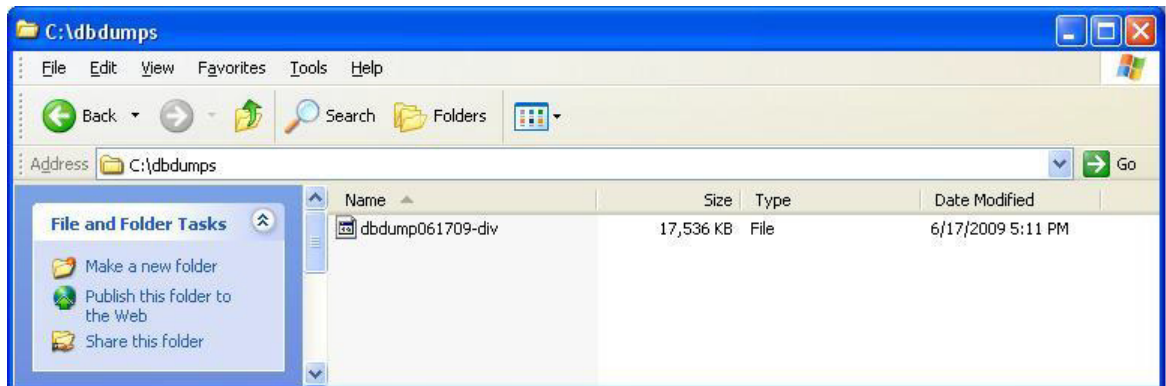
9. (Optional) You can list the directory contents to get the name of the file you want to transfer, by entering the command: `ls -l`
10. To transfer the dump file, enter the get command using the following format:
`get /tmp/<filename>`

```

C:\WINDOWS\system32\cmd.exe - ftp -i
C:\dbdumps>ftp -i
ftp> open 172.24.96.55 49152
Connected to 172.24.96.55.
220 (vsFTPd 2.0.5)
User (172.24.96.55:(none)): so
331 Please specify the password.
Password:
230 Login successful.
ftp> cd /tmp
250 Directory successfully changed.
ftp> binary
200 Switching to Binary mode.
ftp> ls -l
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
--w--x--T   1  0      0              0 Jun 11 20:35 configtab
-rw-r--r--   1 500    501      17039360 Jun 17 23:32 dbdump061709
-rw-r--r--   1 500    501      17956864 Jun 17 23:33 dbdump061709-div
-rw-rw-rw-   1  0      0              13 Jun 11 19:04 iNEWS-wordlength
drwx-----  2  0      0      16384 Apr  2 2008 lost+found
drwx-----  2  0      0       4096 Mar 20 17:58 orbit-root
226 Directory send OK.
ftp: 421 bytes received in 0.00Seconds 421000.00Kbytes/sec.
ftp> get dbdump061709-div
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for dbdump061709-div (17956864 bytes)
226 File send OK.
ftp: 17956864 bytes received in 21.02Seconds 854.48Kbytes/sec.
ftp>
ftp>

```

The FTP command prompt will reappear in the window as a definitive indication that the transfer has finished. However, you can check the directory on the Windows PC for the presence of the file; it is created as soon as the transfer begins, and the size will continue to grow during the transfer process.



The file will be much larger than the one shown in the example here, so the transfer can take a while to complete. Avid suggests conducting file transfers (FTP) from computers that do not have aggressive power saving settings in place.

11. Type **bye** to end the FTP session.
12. Type **exit** to exit the command prompt window.

You are now ready to remove the temporary dump file.



While you can leave one or two in the /tmp folder temporarily, Avid recommends that no more than a couple of database dump files be kept at any given time. Removal of older dump files will help avoid running out of hard drive space on the server.

13. Use the **rm** command at the console to remove old database dump files from the /tmp folder on iNEWS.

- a. Enter superuser mode, first.

```
NRCS-A$ su
```

```
Password:
```

- b. Type the **rm** command, using the following format: **rm /tmp/<file name>**

```
NRCS-A# rm /tmp/dbdump061709-div
```

```
rm: remove regular file `/tmp/dbdump061709-div'?
```

- c. Type **y** to answer verification question.

```
rm: remove regular file `/tmp/dbdump061709-div'? y
```

- d. (Optional) When the prompt reappears, such as NRCS-A#, you can verify the file's removal by viewing the contents of the /tmp folder again.

```
NRCS-A$ cd /tmp
```

```
NRCS-A$ ls -l
```

Backing up Individual Queues

Individual queues can be backed up by specifying the **-n** flag and listing up to 10 queue or directory names:

```
NRCS-A# dbdump c -n <queue or directory name>
```

To back up individual queues or directories to tape:

1. Insert tape into a server's drive and select that server.
2. Use the **dbdump** command in one of the following formats:
 - ▶ 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.2010.FEB** on a new tape, type:

```
dbdump c -n scripts.2010.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.

- ▶ 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 previous example, the next day you would insert the same tape, make sure its write protect is off, and then type **dbdump a** to append that day's scripts to the tape:

For instance, after making the first backup of SCRIPTS.2010.FEB shown in the previous example, back up SCRIPTS.2010.MAR the following month by inserting the same tape and entering the following:

```
dbdump a -n scripts.2010.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 the 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 Linux platform, you get a message similar to the following:

```
NRCS-A# dbdump c
```

```
Do you really want to create a new archive? (y/n) y
```

```
Starting Dump(1) Block (0), dumped on Thur Jul15 `0:36:14: 2010
```

```
dumping isam user
```

```
Please remove tape.
```

```
Insert continuation tape (make sure safety is off)
```

```
When continuation tape is ready type: y (RETURN)
```

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 lighter 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 iNEWS 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:

```
NRCS-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.



When the `-i` option is used with the `dbrestore` command, `dbrestore` does not create any index requests in `SYSTEM.INDEX` when stories are restored to indexed queues. The `dbrestore` process is slowed when attempting to create index requests. Avid recommends the database be restored using the `-i` option and the stories be re-indexed at a later time.

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 restoring 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-temp`. When `dbrestore` has restored all stories in `Scripts-temp`, 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 may not 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.



If the new directory name is longer than the old one, there is a possibility that sub-directory names may become too long and cannot be restored. When the new name is longer than the old name, the dbrestore program will prompt with the following warning message:
WARNING: Alias too long! It should be shorter or the same length as current directory name to avoid exceeding directory name length restrictions.
Do you really want to continue? (n/y)

Listing Tape Contents and Backup Dates

When restoring a directory, you might 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

If you want 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 the following format of dbrestore:

```
dbrestore td -d <date>
```

2. Enter the date in YYMMDD (year, month, day) or YYYYMMDD 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 2009, type:

```
dbrestore td -d 0910
```

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 Jul15 10:30:19 2009
Dir           SHOW.TODAY
Que   27       SHOW.TODAY.SCRIPTS
Que    3       SHOW.TODAY.RUNDOWNS
67 stories listed
```

In the previous example, the tape contains only backups from two dates: the first backup and one made on July 15, 2009. Notice the date of the first backup is not included in the listing. To find the date of the first backup, type 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 Jul10 15:33:01 2009
Continuing with Dump(2) Block(0)
Dumped Thu Jul15 10:30:19 2009
Elapsed time: 00:13
```

In the example, the tape contains two backups, one done on July 10, 2009, and another done on July 15, 2009.

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 82 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 lighter system use. Avid recommends running all tape operations during non-critical periods, not during shows.

Searching a Tape for Stories

To search a tape for stories:

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.

After restoring 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 console. See “[Checking for Free Space on a Database](#)” on page 85 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 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.

Always include the word “for” and one or more words for which you want to search when you use the searchtape command. Even if you use a date or a date range as part of the search criteria, as explained in the next section, you must include a word or words for which you want the system to search.

Use search criteria that is 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

Searching a Tape by Word and Date Range

To restrict a search to stories backed up between certain dates:

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) or YYYYMMDD 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
```



The searchtape command 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, 2010 that contains the words helicopter or president, type:

```
searchtape just 100515 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, 2009 that contains the word helicopter, type:

```
searchtape just 0902 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:

1. Select a server.



For information on selecting a server, see the iNEWS Newsroom Computer System Setup and Configuration Manual.

2. Type: **dbfree**.

A message similar to the following appears:

```
database size 16,386,300 free 3,288,408 = 20%, freelist 3,730
```

In the previous message example the freelist item indicates how many database blocks are used to hold the free list. The freelist is compressed. 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 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:

1. Select the master computer (typically server A).
2. Type: **dbserver 3288508**

The number after dbserver must be the sum of the additional blocks and the free blocks from dbfree.

A message similar to the following appears:

```
NRCS-A Sat Jul24 10:29:26 2010 dbserver complete-exiting
```

At this message, dbserver is finished and you may proceed. You can run dbfree again to verify that you have enough space.

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 continuing: 1500 lines
```

```
Press <del> to skip this story.
```

To skip the large story, press Delete while it is being restored.

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 (or directories) to be backed up:

```
NRCS-A# dbdump ci -d -n <directory or queuename> ..."
```



The -d option dumps the news directory skeleton but will not dump any stories. The -n option will dump the stories in any directories and/or queues specified. The i option will dump the isam files.

The essential directories would be System and possibly Show and a few daily Assignments queues. The System directory must be part of the minimal dbdump.

If master rundown skeletons are stored 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 may list up to 10 directories and queues with the -n option of the dbdump program.

```
NRCS-A# dbdump ci -d -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 is 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.

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 found 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 wireserver 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.

Backing up System Site Files

Your system uses site files in combination with software taken from your release CD and update tapes. Back up your site files on a separate tape. If anything happens to your system, you can use your release CD 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 wireserver to your system requires that you modify your configuration file (/site/config). After making this change, 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:

1. Insert the tape into the master computer (typically server A).
2. Select the master computer.
3. Enter superuser mode.
4. Type: **sitedump**

Linux systems can make sitedumps to their hard drives.

```
sitedump -f /backups/<date>.sitedump
```

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.

7 Disconnects

This chapter provides vital information to help you recover from a system disconnect.

This chapter contains the following main sections:

- [Normal System Status](#)
- [Disconnects](#)
- [Disconnect Recovery](#)

Normal System Status

The displays will vary when the system is a triple system configuration.

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:

```
NRCS-A# status

A is ONLINE and has been CONFIGURED. ID is NRCS.

System is AB. Master is A.

Disk status is OK. The database is OPEN.
```

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:

```
NRCS-B# status

B is ONLINE and has been CONFIGURED. ID is NRCS.

System is AB. Master is A.

Disk status is OK. The database is OPEN.
```

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.

Disconnects

If the servers disconnect from each other, the databases are no longer mirrored between the two servers and immediate attention is required. Since mirroring has stopped, stories written by users on server A will not be seen by users logged in to server B. 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 95](#).

If a disconnect occurs, users may want to export the stories they are working on to local hard drives, in addition to normally saving them to the server. They can export stories by selecting File > Export Story. This will make a backup copy of the story on the local hard 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. The system issues a popup to alert users who must contact their system administrator immediately.

Detecting a Disconnect

You can check to see if the servers are connected to each other and mirroring at any time using the status command at the console. 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.
```

See [“Types of Disconnect” on page 93](#) for more information.

If the servers disconnect, a disconnect warning message appears on the console.

On the Linux platform, the messages include detailed information from the driver (mp) that controls the mirroring:

```
Jul 13 07:45:44 nracs-a IO handler: B silent for 30 seconds
Jul 13 07:45:44 nracs-a IO handler: LINK TO B FAILED, DISCONNECTING B
Jul 13 07:46:05 nracs-a msg: 66 to server on computer A
Jul 13 07:46:25 nracs-a S508: [15028] monitor 508 508 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S510: [15029] monitor 510 510 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S238: [14998] seek 238 238 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S240: [14999] seek 240 240 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S236: [14997] seek 236 236 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S232: [14995] seek 232 232 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S234: [14996] seek 234 234 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S254: [15001] action 254 254 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S504: [15026] monitor 504 504 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S506: [15027] monitor 506 506 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S508: [15028] monitor 508 508 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S510: [15029] monitor 510 510 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S238: [14998] seek 238 238 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S240: [14999] seek 240 240 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S236: [14997] seek 236 236 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S232: [14995] seek 232 232 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S234: [14996] seek 234 234 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S254: [15001] action 254 254 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S258: [15003] action 258 258 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S252: [15000] action 252 252 Server - Hot-to-go
Jul 13 07:46:25 nracs-a S256: [15002] action 256 256 Server - Hot-to-go
```

```
Jul 13 07:46:25 nracs-a disconnect: B COMPUTER DISCONNECTED
```

In addition to console warning messages, a warning message is broadcast to all users currently logged in to the system via a popup window at the workstations.

Types of Disconnect

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:

```
NRCS-A# status
```

```
A is ONLINE and has been CONFIGURED. ID is NRCS.
```

```
System is A. Master is A.
```

```
Disk status is OK. The database is OPEN.
```

The single system status is reported identically on both servers:

```
NRCS-B# status
```

```
B is ONLINE and has been CONFIGURED. ID is NRCS.
```

```
System is B. Master is B.
```

```
Disk status is OK. The database is OPEN.
```

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 RX-ERRs, 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.

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

After 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 be referred to as the failed server.

Before the failed server can be reconnected to the survivor, it must be rebooted and its database wiped clean. After 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, discovering a disconnect as soon as possible 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 95.

The disconnect recovery steps are as follows:

1. Reboot the failed server.
2. Clear the database on the failed server.
3. Reconnect the failed server.
4. Copy the database to the failed server.
5. Log out and stop all servers and resources.
6. Reconfigure the system.
7. Start up all servers and resources.

Procedures

This section contains an example of recovery from a disconnect of a dual AB system. The steps might vary when the system is a triple system configuration. Triple system configuration customers might need to contact Avid for more specific instructions.

In the example, it is assumed that server A was chosen as the survivor and server B was designated the failed server:

- NRCS-A–survivor
- NRCS-B–failed server

The failed server is also referred to as the revived server after 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:

- ▶ Select File > 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.

To recover the system:

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.

2. Halt and reboot the disconnected (failed) server using the following command:

```
NRCS-B# shutdown
```

```
NRCS-B# reboot
```

3. Log in to the rebooted failed server as system operator, then become superuser.

- a. At the login prompt, log in as the system operator user (so):

```
iNEWS Newsroom Computer System
```

```
login: so
```

```
Password:
```

```
Last login: Mon Jul 19 07:17:23 on ttyS0
```

```
?:
```

- b. When you log back in you will be at the question mark prompt because the server is not yet named (connected). You must be in superuser mode for the next step:

```
?: su
```

```
Password:
```

```
?#
```



For more information on superuser mode, selecting servers, and configuring the system, see the [iNEWS Newsroom Computer System Setup and Configuration Manual](#).

4. 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
? Mon Jan  3 16:18:23 2000 Each dot represents 10,000 blocks. The entire
database = 1677 dots.
.....10.....20.....30.....40.....50
.....60.....70.....80.....90.....100
.....110.....
```

The diskclear will print a sequence of dots and numbers as it clears the disk. For instance, on a full 16 gigabyte database, the dots and numbers count to 16,000. The diskclear may take some time to complete.

5. Reconnect the failed server to the survivor.
 - a. 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
```
 - b. Depending on which one failed and which one is master, you would enter one of the following commands:


```
reconnect a master=b net=ab
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.
6. Begin copying the database back over to the failed server.
 - a. Select the revived (failed) server only on the console and begin a diskcopy:


```
NRCS-B# diskcopy -9
```

Users can continue working while the diskcopy continues in the background.
7. When the diskcopy has completed, stop all servers and devices.

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.

Schedule a time to log the users off and stop the servers and devices. At that time, run the following commands on the survivor:

```
NRCS-A# offline
NRCS-A# logout all
NRCS-A# stop all
```

8. 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:

```
NRCS-A# offline
NRCS-A# configure
NRCS-A# online
```

Wait for the system being configured messages to appear on both servers before moving on to the next step.

9. 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:

```
NRCS-B# startup
```

10. 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.

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 want to make server A the master computer again, perform a normal system shutdown, reboot and startup, as described in *iNEWS Newsroom Computer System Setup and Configuration Manual*.



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.

Recovery 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 95.

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.

Computer	A/B	Commands to Run:
survivor	<input type="checkbox"/>	Enter superuser mode.
failed	<input type="checkbox"/>	shutdown reboot Login as system operator. Enter superuser mode. diskclear -
both	AB	reconnect <failed> master=<survivor> net=ab Type either of the following: reconnect a master=b net=ab reconnect b master=a net=ab
failed	<input type="checkbox"/>	diskcopy -9

Computer	A/B	Commands to Run:
survivor	<input type="checkbox"/>	offline logout all stop all configure
both	AB	Wait for the “system being configured” messages on both
survivor	<input type="checkbox"/>	restart all online
failed	<input type="checkbox"/>	startup Be patient for the server to come up.

8 Troubleshooting

This chapter provides information to help you recover from various kinds of system failures.

This chapter contains the following main sections:

- [iNEWS Workstation Problems](#)
- [Wire Problems](#)
- [Locked Blocks](#)
- [How to Check Process Status \(ps Command\)](#)
- [Power Failure](#)
- [Network Failure](#)
- [iNEWS and FTS Connectivity \(ftsdebug Command\)](#)

iNEWS Workstation Problems

This section explains common problems users might have during daily operations. Possible solutions appear after each problem.



A user who installs the client software, iNEWS Workstation, must have Administrator or Power User permission.

A User Cannot Log in

If a user is unable to log in to the iNEWS Workstation and gets the Invalid User Name/Password error message, ensure the user:

- Entered correct user name and password, spelled correctly
- Specified proper server name

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 [“Database Security” on page 39](#) 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:

To check server network connections:

1. Try to ping the server to check network connection to the server, by doing the following.
 - a. Click Start.
 - b. Select Programs > Accessories > Command Prompt to open a DOS window.
The name and location of this submenu option might vary depending on the Windows-based operating system.
 - c. Enter the following command, substituting the name of your server:

```
C:\>ping nrcc-a
```

A message similar to the following should appear:

```
Pinging nracs-a [10.1.38.30] with 32 bytes of data:
```

```
Reply from 10.1.38.30: bytes=32 time<10ms TTL=255
```

```
Reply from 10.1.38.30: bytes=32 time<10ms TTL=255
```

```
Reply from 10.1.38.30: bytes=32 time<10ms TTL=255
```

```
C:\>;
```

If you do not get replies, then the server is either down or a networking problem exists. A brief message, similar to one of the following lines, might appear:

```
Bad IP address nracs-a
```

```
Bad host nracs-a
```

The iNEWS 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.



The workdebug command may be used in system operator or superuser mode.

2. If the workstation can successfully ping the server, the next step is to run the workdebug stat command on each server at the console.

The workserver process runs on iNEWS 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:

```
NRCS-A# workdebug stat
```

```
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 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 Avid 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, select the filter to send commands to all servers via the PuTTYCS application and type:

```
NRCS-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:

```
NRCS-A$ workdebug silent
```

A User Cannot Access an Item

If a user cannot read or write stories in part of the database, he or she may not have permission to do so. Check the user's group permissions.

Group Permissions

To assign permission to a user:

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 the *iNEWS Newsroom Computer System Setup and Configuration Manual* 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 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 might 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 might see messages on the 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 unbusy.

The Dead queue is usually large and an unbusy on the entire queue could take a long time to execute.

To unbusy edit-locked stories in the Dead queue:

1. Enter superuser mode on the console.
2. Use the unbusy command to remove the edit locks.

```
NRCS-A# unbusy -i dead
```

The -i option allows the unbusy command to ignore the inverted attribute.

3. 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) or use the Delete key to "interrupt" the unbusy process.

Wire Problems

This section explains common problems users might encounter with their wire ingests during daily operation.

Data Receiver

The Avid Data Receiver provides an Administration application that may be used to troubleshoot most issues related to ingesting wires to iNEWS using a Data Receiver instance. For more information, see the *Avid Data Receiver Installation and Operations Manual*.

If stories are not being sent to iNEWS—that is, the number of stories processed never increments in the Administration application—ensure raw data logging is enabled and functioning. If it's not, check your hardware connection and the Data Receiver configuration. If some raw data is being logged, but stories are still not being sent to iNEWS, then check the start/stop sequence configuration for Serial and Telnet input types. Also ensure file logging is set to ALL and check the log file for parsing errors. This would indicate a possible problem with parsing NSML, which could make iNEWS reject stories when Data Receiver attempts to send them.

Locked Blocks

During normal operations the system constantly locks and unlocks blocks on the servers' hard drive as data is accessed. Occasionally, something may go wrong and a locked block is left behind.

If you have a persistently locked block, any other process that tries to access the locked block will wait for the block to become available. The usual scenario for this problem is that PCs start hanging in the newsroom when users get to a story in a rundown that is in a locked block. Users then move to another PC, and hang that one too when they access the locked block story.

If the original locked block is unlocked, then all frozen PCs “unhang.”

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 might be a locked block situation. If a block remains locked for more than one minute, the system will print a diagnostic on the console and will repeat the message every minute the block remains locked. The system will remove block locks owned by processes that have exited.

If you have a persistently locked block, contact Avid.

How to Check Process Status (ps Command)

Occasionally, the support staff might ask you to run a few Linux 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.

```
NRCS-A$ 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.

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`:

```
NRCS-A$ 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 might 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, 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:

To connect and start up servers after a simultaneous power failure:

1. On all servers simultaneously, type: **connect <server letter>**
2. To remove edit locks, on the master computer only, type: **dbclean -x .**



This command is not strictly necessary. “Stale” edit locks are detected and ignored. An edit lock is considered to be “stale” if the session that owns the lock no longer exists or if the session was started—logged in—after the edit lock was acquired.

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 might 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.

Network Failure

The iNEWS newsroom computer system 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 is echoed back from the target server and returns to the sending computer in the form of ICMP replies. If the network or server is down, you will not get replies from the server (or computer) you are trying to ping. You can ping a computer using its IP address or name, as shown in the following examples:

```
NRCS-A$ ping -c 5 152.165.17.110
```

```
NRCS-A$ ping -c 5 nrcs-a
```

In the previous ping command examples, the number of ping requests sent would continue endlessly. To control the number of ping requests use a specific number to limit the number of requests. For instance, in the following examples, five ping requests will be attempted as noted by including the number 5 in the ping command.

```
NRCS-A$ ping 5 152.165.17.110
```

```
NRCS-A$ ping 5 nrcs-a
```

In the case of an iNEWS 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. (The syntax for ping on Windows is different than the Linux ping command.)

If computers are unable to ping each other, check for a loose or disconnected network cable or a hub that might 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 (RX-ERRs) have been detected
- How many output errors (TX-ERRs) have occurred

Output Errors (TX-ERRs)

The primary barometer of network health is the output errors (TX-ERRs) column. The computer will try to transmit a packet 15 times before chalking the attempt up as an TX-ERR. It will then try to transmit again 15 times and may wind up incrementing the TX-ERR count. Clean networks will show zero TX-ERRs, or no more than a few.

Of more concern than the raw number of TX-ERRs is how quickly they are increasing. If you are picking up an TX-ERR 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 TX-ERRs your system produces each week or month.

Input Errors (RX-ERRs)

Input errors (RX-ERRs) are fragments of packets or unrecognizable packets. Systems co-existing on a Novell Local Area Network (LAN) generally show many RX-ERRs, although they do not seem to cause problems.


iNEWS and FTS Connectivity (ftsdebug Command)

For more information about Fast Text Search and its utility programs, see the *iNEWS Newsroom Computer System Setup and Configuration Manual*.

The ftsdebug command can be used to verify communication between the iNEWS Servers and the two server (utility) programs related to Fast Text Search (FTS). It can also be used to compare the contents of indexes and indexed queues, adjust the indexes, and is a vital tool for FTS troubleshooting and operation. The format for the command is:

```
ftsdebug [index | search] <parameter>
```

The index option will apply the ftsdebug command to the utility program used by FTS, known as the ftsindex server or ftsidx.exe. The search option will apply the ftsdebug command to the utility program known as the ftsseek server or ftsch.exe.

Parameter	Description
closelog	Close the log file (leaves log intact).
debug	Set debug log level for the program to: Level 0 - no logging done Level 1 - each action logged Level 2 - request parameters logged Level 3 - request parameters and log each action
openlog	Open/reopen the log (truncates existing log).
shutdown	Shutdown the program on the FTS server.  <i>There is no start parameter because the FTS programs must be started from the FTS server.</i>
statistics	Show program version, start time, and statistics.
viewlog	Print the contents of the log file
Limited to index only:	
showrequests	List pending index requests (SYSTEM.INDEX)
tune on off	Turn tuning on or off or force a major tune.
tune <type>	Types of tunes are: minor, major, and absolute (which tunes immediately)

Parameter	Description
Limited to search only:	
adjust [-v] <path>	Compare FTS index for <path> against database content and fix discrepancies. Used to bring indexing up-to-date. For instance, type: ftsdebug index adjust archives.2002
compare [-v] <path>	Compare FTS index for <path> against database content.
list [-n] <path>	List FTS index entries for <path>
search [-v] <query>	Search FTS index for <query> terms

Checking Communication between iNEWS and FTS Servers

Verify communication from the iNEWS Servers to the FTS utility programs running on the Windows-based FTS server by using the following forms of the ftsdebug command:

```
ftsdebug index stat
```

```
ftsdebug search stat
```

These commands may also be used to monitor the indexing process.

After each command is entered at the console, a display of statistics about the utility program, either the ftsindex server (ftsidx.exe) or the ftsseek server (ftssch.exe), appears. The display will vary depending on which form of the command you used. Samples of each are shown in the following two columns:

NRCS-A\$ ftsdebug index stat	NRCS-A\$ ftsdebug s stat
Program name: ftsidx.exe	Program name: ftssch.exe
Version: 3.1.0.2 FTS	Version: 3.1.0.2 FTS
Process id: 1172	Process id: 2024
Started: 06/17/10 09:00:52	Started: 06/17/10 09:00:52
Debug value: 0	Debug value: 0
Tuning: on	Current Role: 1
Index Base: C:\FTS	Index Base: C:\FTS
Current Index: 029	
Indexed Documents: 1386610	Requests Serviced: 57
Requests Serviced: 1669414	
000 File sizes (KB): 499/440 508/1 713 1	
Documents: 1346 Added: 2759 Removed: 1393 Replaced: 15	
Purged: 35	
002 File sizes (KB): 1/0 1/0 2 1	
Documents: 2 Added: 2 Removed: 0 Replaced: 0 Purged: 0	
005 File sizes (KB): 223941/223921 45479/1 402846 1	
Documents: 962661 Added: 962661 Removed: 0 Replaced: 0	
Purged: 0	
029 File sizes (KB): 310330/302993 79098/1 156588 1	
Documents: 356213 Added: 459650 Removed: 61 Replaced: 0	
Purged: 103376	
040 File sizes (KB): 53628/0 147319/0 28849 1	
Documents: 66388 Added: 131984 Removed: 48163 Replaced: 7	
Purged: 17440	

The number of requests serviced includes index requests, purge requests, and ftsdebug status requests, so it is unlikely to match the number of indexed documents.

If communication with the FTS utility programs on the FTS server is unavailable, the ftsdebug command will hang, and then eventually time out and respond with a message that it couldn't connect. If this happens, check the Windows-based FTS server to see whether it rebooted. Also, confirm whether the utility programs (ftsidx.exe and ftssch.exe) are running. Stop and restart them, then check for communication again.



If the ftsdebug command hangs, you can break out of it by pressing the Control and C keys (Ctrl+C).

Community

The iNEWS Community communication protocol requires compatibility between server versions and database sizes, as explained in the *Avid iNEWS Setup and Configuration Guide*. When incompatibility occurs between systems in an iNEWS Community, error messages are provided.

The following table shows error messages and the cases in which they appear.

Case	Error Message
iNEWS version 3.4.5 with large database connecting to version 3.4.3 or earlier	local: G###: Error (1) during SETCOMMUNITY to <REMOTE> remote: C###: Cannot service request: eB
iNEWS version 3.4.3 or earlier connecting to version 3.4.5 with large database	local: G###: Unable to set community server number with <REMOTE> (9) remote: C###: Cannot participate in small database community
iNEWS version 3.4.5 with large database connecting to version 3.4.5 with small database	local: G###: Unable to set community server number with <REMOTE> (9) remote: C###: Cannot participate in large database community
iNEWS version 3.4.5 with small database connecting to version 3.4.5 with large database	local: G###: Unable to set community server number with <REMOTE> (9) remote: C###: Cannot participate in small database community



There was no version 3.4.4 of iNEWS released.

A Command References

Most of your system's commands are special commands provided by Avid. The commands you are most likely to need are listed and explained in this appendix, along with examples.



Some available commands are meant to be used only by Avid technicians or under the supervision of Avid personnel. These commands may cause damage if used improperly. They are listed in this appendix on “[Commands Used by Avid Personnel Only](#)” on [page 115](#).

This appendix contains the following main sections:

- [Programs Invoked by iNEWS](#)
- [Commands Used by Avid Personnel Only](#)
- [Linux Commands Used in iNEWS](#)
- [Console Server Commands](#)
- [Job List Commands](#)
- [Dialog Commands](#)

Programs Invoked by iNEWS

The following programs are invoked and used by your iNEWS system. Do not use them as commands.

action	ftsseek	nxserver
bio.conf	gnews	parallel
bioserver	ismessage	rxnet
boot	keyword	seek
brand	license	server
cgi-fcgi	mailserver	snews
connect.sh	monitor	start

copyright	monserver	txnet
disconnect	news	webaccserver
distribution	newsmail	workserver
ftsindex	NewsWeb.fcgi	

Commands Used by Avid Personnel Only

The following commands are used by Avid personnel only.

attach	dbsize	netterm
bdump	detach	nsupgrade
binhex	diddle	qcheck
biodebug	edit	qstampcheck
biolatency	finit	qxcheck
biosleep	ftsdebug	rcat
biostat	gnewslog	strerror
catchcheck	ifis	sysname
ccuputkey	ifmaster	traverse
clearflagbyte	ifsu	userclean
cmrdebug	keycheck	wordbreaks
dbcoppy	kwcheck	workdebug
dbgen	link	wxlate
dblinks	msgdebug	xi
dblock		

Linux Commands Used in iNEWS

The following Linux commands are available in iNEWS. For more information, see the reference material that came with your Linux 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`

cat	kill	pwd
cp	more	rm
date	mv	sync
df	passwd	telnet
grep	ps	

Console Server Commands

You must enter commands in lowercase. Your system does not recognize commands entered in uppercase.

broadcast

```
broadcast [-dl | --] <message>
```

Sends a message to everyone logged in. The `-d` option will issue the broadcast in a popup window. The `-l` option will issue the broadcast only to those directly logged in to the local system, not through Community. For instance, to send a message, select one server and type:

```
NRCS-A$ broadcast -d System going down at 12:00
```



*Certain characters are interpreted by the bash shell program, so when including characters, such as angled brackets, exclamation points, asterisks, or pound signs (<, >, !, *, #) in the <message>, the entire message should be placed within quotation marks to prevent the program from interpreting them.*

Use the double-hyphen (`--`) option when message begins with a hyphen (`-`).

configure

```
configure [-ns] [<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 rxnet 310 and 311, which are connected to server A in an AB system. To test these changes, become a superuser and type:

```
NRCS-A# configure /site/config ab a
```

If no <config file> is specified, /site/config is used.

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.CLIENT.WINDOWS directory, 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.

connect

```
connect <name> [<option>=<value>] ...
```

The connect command names each server in the system—one of A, B, C, or D—and tells each how many other servers there are in the system and how to communicate with them. For instance:

```
NRCS-A$ connect a net=ab
```



The connect command reads the /site/system file for options. If an option is specified on the command line and also in the /site/system file, the command line setting takes precedence.

connect Command Options

auto_upgrade=<yes no>	msgserver=<silent verbose>
clockmax=(12 24)	name=<a b c d>
disk=<status>	net=<name,name[,name]>
excludedvideo=(director none)	pausetimeout=<mm:ss>
highwater=<number_of_blocks>	purgelimit=<number of hours>
id=<system name>	readrate=<number of words per minute>
lastlogin=<yes no>	remotetimeout=<mm:ss>
load=<number>	security=<and or>
localtimeout=<mm:ss>	single=<name> or net=<name,name[,name,name]>
lowwater=<number_of_blocks>	timechar=<character>
master=<a b c d>	timer=<silent verbose>

connect Command Options

maxhits=<number>	useclienttimezone=(yes no)
min_passwd_length=<number>	wordlength=<number>



The useclienttimezone option is used with date search options.

ctraits

```
ctraits <system name> [<option> <value>]... [+flag]... [-flag]...
```

The ctraits command adds systems to a Community and defines the order in which a local system will contact a remote system’s server in the Community, as well as membership, messaging and connections.

You must be in superuser mode at the console to use the ctraits command. For instance:

```
NRCS# ctraits WNWS suffix AB +m +o +i
```

Option or Flag	Description
suffix <list>	The required suffix option defines the order in which the local system will contact the remote system's servers. The list is the combination of ABCD in the preferred order. For instance: <code>ctraits suffix BA</code> would first contact the remote system's B server, and then the A if B was unreachable.
group <group>	(Optional) Defines the list of users who are permitted to access the remote system when connected to the local system. Superusers on the local system will see all available Community members, regardless of group definitions.
remove (no value)	Remove a system from a Community. For example, type <code>ctraits WNWS remove</code> to remove WNWS from WAVD’s Community.
+m -m	This flag is required to allow messaging to the target server.
+o -o	This flag is required to allow outgoing connections to the target server. All +o systems appear in the Directory panel for permitted users.
+i -i	This flag is required to allow incoming connections from the target server.

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 or log everyone off the system by typing `logout all` before issuing this command.

Use the double-hyphen option (`--`) when `<directory pathname>` begins with hyphen (`-`).

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:

```
NRCS-A$ logout all
```

```
NRCS-A$ dbclean -x .
```



Without the `-x` option, no database changes are done.

dbclose

```
dbclose
```

Closes the database.



If you use this command while users are active, changes to stories will be lost.

dbdev and dbsize

```
dbdev
```

Reports the number of 4k blocks allocated for the `/dev/rp5` partition. There can be only a single partition for the database.

```
dbsize
```

Reports the size of the database. The size of the database can be smaller than the size of the partition.

To find out the size of your partition, type: **dbdev**

A message similar to the following appears

```
/dev/rp5 size is 39,319,078
```

To view the size of the database, type: **dbsize**

A message similar to the following appears

```
Database size is 33,554,431
```

dbdump

```
dbdump <keys> [<option>] ...
```

Dumps individual stories or the entire database to file, network, or tape drive.

This command can be interrupted. The program will continue dumping to reach an appropriate quitting point.

Valid Keys	Description
a	Append to current dump
c	Create new dump
C	Create new dump, don't ask if it's ok.
d	Dump the news directory
i	Dump indexed files (such as, user index)
p	Dump projects
s	Show quick index of dumped stories
v	Verbose output
x	Ask before dumping indexed file

Valid Options	Description
-a <device>	Use alternate device for dump
-d	Dump news directory skeleton (no stories)
-f [file]	Dump to file (use '+' for standard output)
-m <minutes>	Dump files modified in last <i>x</i> minutes
-n <namelist>	Only dump listed directories
-N <computer name>	Network dump to specified computer
-P <namelist>	Dump specified projects
-X	Dump skipped queues

To dump everything, except those directories marked with a skip flag, type: **dbdump c**

To include directories/queues with skip flag on the dump content, use the `-X` option:

```
dbdump c -X
```



When neither the `p` nor `d` key are specified with the command, such as when `dbdump c` command is entered, both directories and projects are dumped.

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
```

Dumps already on the tape are skipped and this dump is added to the end.



This command does not skip the contents of all subdirectories and queues if only the parent directory has the skip trait enabled; contents will only be skipped by `dbdump` if the actual subdirectory or queue has the skip flag.

dbfree

```
dbfree [-cf]
```

Reports the size of the database and the size of the free list—that is, the amount of free blocks available. The size of a database block is 4k.

To display this information, type: **dbfree**

A message similar to the following appears:

```
data base size 33,554,431 free 33,400,000=100%, freelist 3,340
```



The freelist is compressed, so there can be as many as 10,000 free blocks accounted for in each freelist block.

To display the amount of free space in your software area, type: **df**

Prior to the system being configured, there are two maintenance options that may be used with the `dbfree` command.

The format is `dbfree [-cf]`.

The `c` option is used to check for cleared free blocks. The `f` option is used to “unclear” cleared free blocks.

dblines

```
dblines [b | c | +D | f | n | -P | q | s | v | w |-O] <pathname>
```

Checks the database for story errors.

Valid Keys	Description
b	Story block count only (no checking)
c	Complete check
+D	Fix future dated queue entries
f	Compare fields in story form with fields in story. Report those not found in both.
n	Do not fix errors
-O	Do not report orphaned story record errors
-P	Do not report “has projects” errors
q	List queue names
s	Skip queues that are skipped by dbdump
v	Verbose output
w	Check word counts and read rate
x	For qstamp checking



The dblines program normally will not fix script related errors; if you want to do so, the +S option must be specified and the n option must not be specified.

Use this command weekly as part of normal database maintenance. Start dblines before you go home and run it in the background. Including a period with the command checks the entire database.

If dblines finds any errors related to queue corruption, call Avid for assistance.

dboriginal

```
dboriginal [-a | --] <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.

Use the double-hyphen (**--**) option when `<pathname>` begins with a hyphen (**-**).

The **dboriginal** command will print diagnostics, indicating how many stories were examined and how many old versions were removed. The command removes old versions for unshared stories—those with a link count of 1—in a queue. The option, **-a**, allows the command to remove old versions from all stories (shared and unshared).



*The **dboriginal** command can also be used to remove old versions from stories in a queue that has the **save none** attribute. When stories are moved to or copied to a queue, all old versions are retained even if the queue has this attribute; only when a story is edited are the old versions removed.*

As indicated by the pound sign (**#**) in the prompt, you must be in superuser mode to execute commands. Also, you must use 24-hour military time.

It should only be necessary to run this command once or twice a year. Below is an example of using the **at** command to execute a **dboriginal** command at a specified schedule and directing it to a specified area (**archive**).

```
NRCS-A# at 11:10
/exc/dboriginal archive
<Ctrl+D>
job 1001340809.a at Sat Jul 10 11:10:00 2004
NRCS-A#
```

dbpurge (Superuser conditional)

```
dbpurge <path> [- h | l | f] [<interval>]
```

dbpurge purges the database to regain space.

Valid Keys	Description
-	Condition used as <code><path></code> parameter. Purges all queues with the default interval

Valid Keys	Description
-v	Condition used as <path> parameter. Purge all queues listed in the command in verbose mode
-h	Include held entries (must be superuser to use this option) in the purge
-l	Include locked entries (must be superuser to use this option) in the purge
-f	Include future dated entries (must be superuser to use this option) in the purge Interval is expressed in <hours> or <days>.<hours>



Any queues that have a write group assigned will not be purged when dbpurge runs as a non-superuser.

The system automatically purges the directories and queues each hour to move old material from high-turnover queues (such as the Wires queues) into the Dead queue. Normally locked and held stories are not purged regardless of their age. 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 queues older than five hours, type: **dbpurge wires 5**

Typing `dbpurge` - purges all queues in the database according to each queue's purge interval. `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] ...
```

The `dbrestore` command restores data dumped via `dbdump` to the iNEWS database. Automatic data conversions are performed to convert data from older database revisions to the current database format.

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.

dbrestore Command Key Options

Keys:

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 each title
x	Ask before restoring each ISAM file
p	Restore projects files
f	Print facts about blocks and times
t	Print table of contents; do not restore

Options:

-a <device>	Restore from alternate device
-c <filename>	Database conversion profile filename
-CCS <name>	When restoring non-Unicode dbdumps (pre-2.5)
-d	<date>[<-date>]
-f [file]	Restore from a file (use '+' for standard input)
-i	Do not index. Do not post index request to SYSTEM.INDEX when a story is restored in an indexed queue.
-k	<keyword> (You can specify multiple keywords.)
-L	<size>[:<repeat>] (large story diagnostic - default 1000:1500)
-m <value>	Maximum number of stories to restore
-M	Preserve modification times
-n [=]	<directory>[=<new name>]



Use c instead of =, as in -n[c], when directory or olddir contains an equal sign (=).

dbrestore Command Key Options

-N	Read from network socket for dump data
-p <queue>	Only with key letter s, will create queue
-P <project-uuid>	Restoring a specified project
-s <platform>	sgi, mip, or sco



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.

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.

When you use dbserver, specify the total number of free blocks you want to have in the free list. If the free list contains 25,000 blocks and you want to build up the free list to 100,000 blocks, type: **dbserver 100000**



When the free list contains the desired number of blocks dbserver stops. If you specify a number that is smaller than the current free list size, dbserver will not do anything.

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.

The dbserver command is invoked when the system is booted and runs in the background continually monitoring the number of free database blocks based on the high and low water marks specified in the system profile.

dbsort

```
dbsort [-v] <queue name>|<project uuid>
```



You can use a period (.) as a wildcard for all queues or as a wildcard for all projects.

The dbsort command sorts stories in any queue by the quick-index field, and should primarily be used to verify that the quick-index field accurately reflects the story sort field contents. For instance, if RUNDOWN.AM has the page-number field set as its quick-index field, type the following to sort the queue by page number:

```
dbsort rundown.am
```

If no `quick-index` 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.



Dbtraits will automatically sort a queue when the sort attribute is turned on with the `+so` option. An ordered queue is automatically sorted when the ordered attribute is turned off with the `-o` option.

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 `quick-index` field so that your next sort is based on current information.

dbtraits

```
dbtraits <pathname> [only] [<option> <value>]
[+|- mode] ...:
```

dbtraits Command Options

<code>changeform</code> or <code>cform</code>	<code>queueform</code> or <code>qform</code>
<code>displaylines</code>	<code>readgroup</code> or <code>rg</code>
<code>editorialgroup</code> or <code>eg</code>	<code>reindex</code> (<code>all</code> <code>media</code> <code>fts</code>)
<code>ftsindex</code>	<code>save</code> (<code>-</code> <code>last</code> <code>orig</code> <code>none</code> <code>all</code>)
<code>mailbox</code> or <code>mb</code>	<code>sortfield</code> or <code>sf</code>
<code>notify</code> or <code>ng</code>	<code>storyform</code> or <code>sform</code>
<code>purgeinterval</code> <days.hours>	<code>stripform</code>
	<code>writegroup</code> or <code>wg</code>

Sets and modifies database traits.

To assign a story form called `rundown` to the `RUNDOWNS.5PM` directory, type:

```
dbtraits rundowns.5pm storyform rundown
```

To assign a queue form called `rundown` to the `RUNDOWNS.5PM` directory, type:

```
dbtraits rundowns.5pm queueform rundown
```



Assigning forms can be combined to one command line, such as:
dbtraits rundowns.5pm storyform rundown queueform rundown

The `ftsindex` attribute supports values from 0 to 49. To assign an index base of 49 to the `WIRES.ALL` queue, type:

```
dbtraits wires.all ftsindex 49
```



All directories default to index base 0. The `ftsindex` attribute is independent of the `INDEXED` database trait and determines which FTS index base is used for the data in the queue with that `ftsindex` value.

dbtraits Command Modes

<code>f</code>	Forms allowed
<code>g</code>	General
<code>i</code>	Inverted
<code>index</code>	Indexed (FTS)
<code>mi</code>	Media Index
<code>o</code>	unordered (- only)
<code>p</code>	Queue print allowed
<code>q</code>	Queue operations allowed
<code>r</code>	Read access
<code>refresh</code>	Queue refresh
<code>s</code>	Sequential
<code>so</code>	Sorted
<code>t</code>	Text timing clocks
<code>u</code>	Update
<code>w</code>	Watch appends
<code>x</code>	Skip

dbvisit

```
dbvisit -<d | v | i> -[r or -c] [-s] [-l] [block# ...]
```


dbvisit Command Options

-c	Use checkpoint partition. Forces a -s to operate in slow mode. This is for online dbvisit.
-d	Display progress by printing dots.
-i	Validate ISAM file record and key sizes.
-l	List link count distribution. The link count distribution is only output when -l is specified.
-r	Read only; do not rebuild free list.
-s	Operate in slow mode to eliminate cache usage
-v	Verbose output; print name of each queue



The validation of ISAM key and record sizes is to prevent problems after software upgrades when the key and/or record sizes of any ISAM files changed for the new software release.

The dbvisit scans the database for errors, then rebuilds the free list and fixes bad story link counts. A list of block numbers can be specified, as shown by the [block# ...] parameter above. A diagnostic is printed whenever dbvisit encounters one of the specified blocks, which is helpful for tracking down database corruption.

In order to use the -c option for an online dbvisit, the checkpoint partition should be active. This is done by briefly shutting down the system, doing a **checkpoint on**, and then bringing the system back online.

Use dbvisit once a month as a part of your regular maintenance.



If dbvisit reports any errors, do not rebuild the free list; call Avid for assistance.

The dbvisit uses asterisks to display information related to search queues as opposed to other queues or queue entries:

- Each dot = 1 queue
- Each asterisk = 1 search queue
- Each colon = 1000 queue entries

For instance:

```
NRCS-A# dbvisit -cd
-----removed for documentation brevity-----
14:57:22 2013-06-30 Using checkpoint partition
14:57:22 traversing roots
-----truncated for documentation brevity-----
14:57:24 traversing directory
14:57:24 (each dot = 1 queue; each asterisk = 1 search queue; each colon =
1,000 queue entries)
14:57:24 *......:.....:10:.....:20:.....:30:.....:40::
14:57:50 :.....:50:.....:60:.....:70:.....:80:.....:90::
-----removed for documentation brevity-----
```

The dbvisit sends its output to the screen that you're using when you run the command. Its output is not recorded by syslog to the iNEWS log file. However, there is a variation on the command you can run to log the dbvisit output file at the same time it displays the output on screen. This provides a record of the dbvisit for later review.

For instance:

```
NRCS-A# dbvisit -cd 2>&1 | tee /home/so/dbvisit_online.log
```

dictionary

```
dictionary [-update] <dictionary> | <directory> ...
```

Any combination of dictionary names and dictionary directories can be specified. For each directory specified, all dictionaries within that directory will be compared to their standard dictionary counterpart. Only missing and obsolete dictionary tokens are identified; definitions are not compared.

The standard dictionary name begins with a pound sign (#) and is expected to reside in the same directory. A dictionary update is only done when the -update flag is specified; otherwise, the dictionary is only checked, by default. Any tokens found in the standard dictionary that are missing from the dictionary will be added and those found in the dictionary that are missing from the standard dictionary will be removed.

diskclear (Superuser only)

```
diskclear [- | -- | ++ | -+u | 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**

Arguments	Description
-	Clear database.
--	Only read database with no changes. Must be superuser.
-+	Clear database; destroy content (Undo cannot restore database).
-+u	Clear, destroy content, and undo (only for install & upgrade).
u	Undo a prior clear database.

diskcopy

```
diskcopy -[n] (Copy Entire Disk)
```

```
diskcopy <start block> [<end block>]!
```

Copies database from master computer (usually server A) to a replacement computer. Enter it on the replacement server. The n represents the number of simultaneous copies per disk. Maximum is 10. Default is 1.

doc

To get text from database use:

```
doc -gt[u] <queue> [slug...] or: doc -gts[u] <queue> <story id>
```

To get NSML from database use:

```
doc -g[2|3] <queue> [slug...] or: doc -gs[2|3] <queue> <story id>
```

The number 2 is for NSML 2; 3 is for NSML 3. If no NSML version is specified, NSML 3.1 is used. If no slugs are given, the entire queue is done. The slug must be one word or enclosed in quotes.

To put text into the database use: `doc -pt[u] <queue> [file...]`

To put NSML into the database use: `doc -p[2|3] <queue> [file...]`

The numbers represent the version of NSML—version 2 or 3. If neither is specified, NSML 3.1 is used.

If no files are given, standard input is read.

The `-g` option does not take any file name parameters; standard output is assumed. If output is to be written into a file, shell file redirection must be used (as in “> output-file-name”). The `-g` option defaults to NSML format input / output and lines are terminated with CR/LF characters.

The `t` modifier specifies that the input / output will be in plain text format. The `u` option specifies Linux line termination is used—that is, lines are terminated with an LF character.

The `-p` option will take input from its standard input if no files are specified. This will put a single story into the database composed of the data read from its standard input up to an end of file condition.

ed

```
ed <file pathname>
```

This command initiates the line editor used to edit 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 are covered in “[The Line Editor, ed](#)” on page 291. The line editor is a Linux-supplied program; refer to Linux documentation for more information.

enter

This command creates directories or queues.

```
enter -d[irectory] | -q[ueue] | -s[earchqueue] <pathname>
```

Add the specified directory, queue, or searchqueue to the news directory.

```
enter -m[ail] [-f[ix]] <username> (can include trailing *)
```

Check that the specified users’ mail queues exist and optionally add the ones that are missing.

force (Superuser only)

```
force -
force [-q] [<name>] ...
force [-q] "created>date1<date2" [<name>] ...
force [-q] "lastlog>date1<date2" [<name>] ...
force [-q] "passchg>date1<date2" [<name>] ...
```



When using angled brackets (>, <) in a parameter, they must be enclosed within quotation marks. The same applies to characters, such as the exclamation point (!) and asterisk ().*

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 iNEWS.

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 (Superuser only)

```
gtraits add <group name>
```

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.

To list group names only, use `gtraits list-`.

To list group members, use `gtraits list [<group name>|<user name>]`.

help

```
help <command name>
```

Displays information on how to use other commands. For instance, to get instructions for the `unbusy` command, type: **help unbusy**

A message similar to the following appears:

```
usage: unbusy [-i][-v][-e][--]<queue name> | <project uuid>
    use "-i" to ignore the inverted attribute.
    use "-v" to show the entire story (implies -e.)
    use "-e" to show empty fields.
    use "--" when <queue name> begins with '-'.
```

hogs

```
hogs [-b] [-v] [--] [<pathname>]
```

Scans the directories or queues you specify and displays usage information for them.

hogs Command Options

-b	Display version block count information.
-v	Display story version count information.
--	Allows next parameter to be <pathname> if it starts with dash (-). The <pathname> can be a project UUID or wildcard semicolon (;) for all projects—as it can be a period (.) for all queues.

You can use this command to get an idea of which queues are consuming the most space and how much is being consumed. This command is most useful when used on the People directory.

For instance, to display usage information for the People directory, type: **hogs people**

Below is an example of using the at command to execute a hogs command at a specified schedule, sending the results to a file (hogs.report), and then using the doc command to copy that file to a queue for later review.

```
NRCS-A# at 11:10
/exc/hogs . > /tmp/hogs.report
doc -ptu people.sysadmin.notes /tmp/hogs.report
<Ctrl+D>
job 1001340698.a at Sat Jul 10 11:10:00 2004
NRCS-A#
```

As indicated by the pound sign (#) in the prompt, you must be in superuser mode to execute commands. Also, you must use 24-hour time.

idiff

```
idiff <file1> <file2>
```

This command allows you interactively compare two files and select and/or edit the portions of the files that differ and create a composite file. The program displays the differing portions of two files and gives you the option to put the portion from the first file or the second file into the output file.

You can also edit the display using the vi program and put the resulting portion into the output file, named idiff.out.

list

list Community	[<name>...]
list [options] configuration [--]	[<termid> <name>...]
list [options] directory [- (f g i o s u v)]	[<name>...]
list group	[<name {name} - +>]
list project [-u]	[<uuid name>]
list [options] queue [- (a b d f g m s v x)]	[<name uuid> [<record limit>]]
list session [- (p v V)]	[<termid> <name> ...]
list sq [-v]	[<name>]
list [options] user [- (b h l m r t v)]	[<name groupname>...]
list Blob [-v]	[<name> ...]

Configuration Options

computer=<name>

mailbox=<number | name>

type=<name>

Directory Options

flags=<flags>



The flags=<flags> directory option allows the use of directory flags from the set SRP(p/l)opLIsUGQ(O/N/A)XWFiTMC to be used to select the directory entries to be listed. To see all indexed directory entries use: list flags=i d. To see all sequential, read-only, and printable directory entries use: list flags=SRP d.

form=<name>



The form=<name> directory option matches the directory entry queue form and story form names.

fts=<index>

ip=<index>

(eg | ng | rg | wg |rwg | mwg)=<group>
mailbox=<number | name>
purge=(<days>.<hours> | <hours>)
save=(last | original | none | all)
sortfield=<field>

-f form

-g groups

-i interplay/fts

-o order-user

-s sons

-u lock-user

-v verbose

Project Options

-u uuids
-s sysid

Queue Options

qindex=<index value>

-a address

-b backwards

-d deleted

-f form

-g groups

-m modified by

-s stamp

-v verbose

-x extended data (distribution)

Search Queue Modifiers

-v verbose

Session Modifiers

-p process id

-(v|V) workstation IP address (V - don't resolve)

User Options

blacklist=(b | -)

external=(E | -)

keyboard=<number>

localonly=(l | -)

password=

readrate=<number>

session=<number>

simplified=(S | -)

su=(n | -)

created>date1<date2

lastlog>date1<date2

passchg>date1<date2

-b blob

-h home

-l last login

-m mail

-r real name

-t management information

-v verbose



The equal sign (=) shown for options represents one of any of these signs (=, >, <, or ~) for equality, greater than, less than, or inequality respectively.

In the option syntax—<option>=<value>—the equal sign (=) can be as described at the end of the usage message to get other relationships, such as `fts~0`, meaning all directory records with an FTS value not equal to zero.

list B

Lists details of the “system blobs” stored on the server. System blobs are used to store bitmaps for the title-entry tool and for the simplified user interface settings.

The following is a sample output:

```
# list B

File-Id      Size  Date      Time      Name
-----
   449      7287  Jul 10 00 07:27:19  BM000001
   457      7238  Jul 10 00 07:28:16  BM000002
  84435      6066  Nov 30 00 08:16:39  BM000003
  84363      6066  Nov 30 00 10:44:13  BM000004
  84403      1171  Nov 30 00 10:56:40  BM000005
  88956  265255  Jan  4 01 16:05:18  BM000006
103749  15439  Mar 22 01 00:40:24  BM000007
   498      11212  Jan 14 00 15:35:28  BM000008
   530      15439  Jan 14 00 15:53:46  BM000009
   546      91416  Jan 26 00 09:47:30  BM000010
   638      14342  Apr  4 00 13:11:17  BM000011
   653       8608  Apr  4 00 13:18:09  BM000012
   662      26673  Apr  5 00 13:46:54  BM000013
104423        40  Mar 21 01 15:16:28  SimplifiedUISettings
```

list C

Lists Community configuration information, including system names and server suffixes.

```
NRCS-A$ list C
```

```

SYSTEM      SUFFIX  moi  GROUP
NRCSWX      AB      -oi  -
NRCS-UK     AB      moi  -
TESTDB      A       moi  sysops
```

```
NRCS-A$
```

list C Information	Description
SUFFIX	The suffix is the order in which the remote systems are contacted.
m	Messaging is enabled.
o	Outgoing is enabled.
i	Incoming is enabled.
GROUP	The group is the group name allowed to see the associated server.

list c

Lists current configuration of the system.

list c Command Options

```
computer=<name>
```

```
mailbox=[<number> or <name>]
```

list d

```
list [<option>] d-[f | g | o | s | u | v] [<directory name>...]
```

list d Command Options

d-f	Queue and story form names
-----	----------------------------

list d Command Options

d-g	Group information
d-i	ftsindex and Interplay information
d-o	Order user
d-s	Son count and sequence number
d-u	Lock user
d-v	Verbose mode

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
SRPl0-LIsUGQSXWFiTmM sortfield  purge dis mbox directory
Q-R-----I--G--X----- TITLE          P3.0  D1  -   DEAD

# list d-f dead
SRPl0-LIsUGQSXWFiTmM queue form story form directory
Q-R-----I--G--X----- DEAD

# list d-v dead
SRPl0-LIsUGQSXWFiTmM 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=
```

For search queues, this command displays an S in the first column and the name of the search queue in the final column, such as:

```
# list d news.football

SRPlo-LIsUGQsXWFiTmM sortfield      purge dis mbox directory
S-----I----- TITLE                P3.0 D1 -      !NEWS.FOOTBALL
```

list g

```
list g [<user or group name>] ...
```

Lists group information.

list p

```
list p [-u] [<uuid> | <name>]
list project [-u] [<uuid> | <name>]
```

Lists projects and facets.

list p Command Options

sysid=<sysid>

- s System ID
- u Universally Unique Identifiers (UUIDs)

 *A project has a 36-character UUID. A facet's UUID is the parent project's UUID followed by a period and another 36-character UUID.*

NRCS-A# **list p -u 161a2017-5cb9-4d41-bb51-9494d4151e5d**

	start	expire	rg	name
P	-	-	-	Elections
F	-	-	-	Governor's Race
F	-	-	-	US HOUSE Races
F	-	-	-	US Senate Race
F	-	-	-	US President's Race
F	-	-	-	State Legislature Races
F	-	-	-	REFERENDUMS

Projects are indicated by a P, while F indicates a facet, and a question mark represents projects or facets that belong to other systems and therefore cannot be used by the current system.

Here is another example of the list p command:

NRCS-A# **list p**

start	expire	rg	name
? -	-	-	BP Payout
? -	-	-	Gulf Businesses
P -	-	-	CRIME
F -	-	-	MOST WANTED
F -	-	-	Local
F -	-	-	STATE
F -	-	-	Federal
P -	-	-	EDUCATION
F -	-	-	COLLEGES & UNIVERSITIES
F -	-	-	SCHOOL BOARD
F -	-	-	ELEMENTARY or HIGH SCHOOL
F -	-	-	HOME SCHOOLING
P -	-	-	ENVIRONMENTAL
P -	-	-	Financial
F -	-	-	Holiday Sales
F -	-	-	Stock market
F -	-	-	Taxes
F -	-	-	Consumer Tips
F -	-	-	Recalls
F -	-	-	Jobs - Unemployment
P -	-	-	Health and Medical
P -	-	-	SWEEPS WEEK

list q

```
list [<option>] q-[a | b | d | f | g | m | s | v |x] <name> [<limit>]
```

list q Command Options

q-a	Record address
q-b	Reverse order
q-d	Include deleted entries
q-f	Story form
q-g	Read-and-write group information
q-m	Who moved, duplicated, or killed the queue
q-s	Queue stamp
q-v	Verbose output
q-x	Extended data (distribution)

Lists information on the contents of a queue. For instance:

```
NRCS-B$ list q people.palmer.new 1
```

A display similar to the following appears:

```
PEOPLE.PALMER.NEW id=126126
rec quick index LHDM-WObfPRmFg 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 -[p | v] [<session id> | <name>]...
```

Lists session information, such as users currently logged in.

The -p option causes the process id for the controlling session process to be printed.

The `-v` option causes the IP address for the session to be printed if the session is a network session, such as an iNEWS Workstation. The `V` option is the same as the `v` option, except it does not resolve IP addresses into names.

list sq

```
list sq [-v] [<search queue name>]
```

Lists the specified search queue's query ID. The `-v` option displays the search queue's query as well. For instance:

```
NRCS-A$ list sq-v news.football
NEWS.FOOTBALL query id:10522758
{MaxFound=100[NRCS]WIRES.ALL}((football))
```

The example shows the NEWS.FOOTBALL search queue's query of the WIRES.ALL indexed queue on the NRCS system for the word football, with the Max Hits set to 100.

list u

```
list [<option>] u[-(h | l | m | p | t | r | v)] [<user or group name>]...
```

Lists user traits information, such as 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:

```
NRCS-A$ list u-v danielmi
```

The verbose result of the command will look something like this:

```
user      rr  kb  su  m  SOEKC VHP  sc  queues
danielmi  180  0   n   i  -OEKC VHP  sc  dest: PEOPLE.D.DANIELMI.NOTES
                                     home: PEOPLE.D.DANIELMI
                                     mail: PEOPLE.D.DANIELMI.MAIL
```

```
NRCS-A$
```

The flags (`rr kb su m SOEKC VHP sc`) in the header provide current status information. The flags are:

rr	Readrate		K	Can Kill All
----	----------	--	---	--------------

kb	Keyboard	C	Can Connect
su	Superuser	V	Can Video Browse
m	Insert/Overstrike Mode	H	Can Highlight Read
S	Simplified User	P	Can Manage Projects
O	Can Order	s	Can Configure Shortcut Toolbar
E	Can Enter & Remove	c	Can Configure Colors

The letters in the header are defined as follows:

u-b	User blobs
u-h	Home queue
u-l	Last login
u-m	Mail queue
u-r	Real name
u-t	Tracking information
u-v	Verbose output
<option>	list blacklist=[b -] user
	list "created>date1<date2" user
	list external=(E -)
	list keyboard=<number> user
	list "lastlog>date1<date2" user
	list "passchg>date1<date2" user
	list password= user
	list readrate=<number> user
	list session=<number> user
	list simplified=[s -] user
	list su=[n -] user



When using angled brackets (>, <) in an option, as shown above, they must be enclosed within quotation marks. The same applies to characters, such as the exclamation point (!) and asterisk (), as shown below.*

```
list user "a*"
```

Lists all users whose login names begin with the letter ‘a’.

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.

makemontab

Installs translations for monitor-related files.

You must be in superuser mode at the console. For instance:

```
NRCS-A$ makemontab -i [sv]
```

makemontab Command Options

s	Standard. Use default translations.
v	Verbose. Show progress.
vv	Very verbose. Show progress plus translations.

makeshift (Super user only)

```
makeshift -[v|i|a|p|r|f <shift-file>] file1 file2 ...
```



This console server command is largely obsolete on unicode systems.

Manages the case-shifting dictionary that iNEWS 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 iNEWS to implement the case-shifting dictionary appropriate for the national language used at your site.

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).
r	Recursive. (For directories.)
<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 -i[sv]
```

maketab Command Options

s	Standard translations
---	-----------------------

maketab Command Options

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, become a superuser and type: **maketab -i**



For maketab, the dictionary definitions are installed into the following programs: /exc/news, /exc/gnews, /exc/snews, /exc/nxserver, and /exc/newsmail. Day and month abbreviation definitions are installed into: /exc/news, /exc/gnews, /exc/snews, /exc/nxserver, /exc/newsmail, /exc/rxnet, and /exc/server.

mapcheck

```
mapcheck [-v] [-f] <queue name>
```

The mapcheck utility validates all stories in a designated queue. Errors are produced for any syntax violations, missing queues (rundown, composite, or event), and trait omissions. Can be started by assigning the system mailbox “map” to SYSTEM.MAP. Sets monitor attribute when missing.

mapcheck Command Options

f	Fix monitored trait
v	Verbose output
vv	Very verbose
vvv	Very very verbose

For example, typing **mapcheck -f system.map** at the console will make the system analyze all stories in SYSTEM.MAP. It sets the monitored trait for any listed rundowns that do not have the trait set, and it reports any syntax errors in SYSTEM.MAP. Mapcheck should be run after an upgrade and as part of error investigation.

msgclean

```
msgclean-<options> [-f <from name>] [-a <age in days>] [-x] [<username>]
```

msgclean Command Options

r	Remove messages
d	Show messages that are removed
o	Show outstanding messages
t	Tabulate outstanding messages
f	Restrict messages to those from the specified sender. Sender is used exactly as specified, no case shift. An empty sender can be specified with quotes ("").
a	Restrict messages to those older than the specified number of days.
x	Restrict to nonexistent users.
<username>	For specified user. Use trailing asterisk (*) for wildcard match. If no username is specified, then an asterisk will be assumed.



When using characters, such as angled brackets (>, <), the pound sign (#), exclamation point (!) and asterisk (), they must be enclosed within quotation marks.*

Type **help msgclean** at the console to view msgclean options at any time.

Removing Messages Sent to Existing or Non-Existing Users

The msgclean command removes messages to existing users as well as non-existing users. When the -r option is used, msgclean prompts for approval for each existing user whose messages are going to be removed.

```
NRCS-A$ msgclean -t west
```

```
west      7  messages
```

```
7 total qualifying messages.
```

```
NRCS-A$ msgclean -r west
```

```
Do you wish to remove messages sent to west (y/n)? y
```

```
7 messages removed.
```

If you remove any users who have unread messages, those messages remain in the database. Use the `-x` option to force `msgclean` to act only on usernames that no longer exist. Messages removed from non-existent users do not prompt for confirmation. The following example shows deletion of three messages pending to a guest user.

```
NRCS-A# msgclean -tx
0 total qualifying messages.
NRCS-A# utraits guest remove
User guest removed
NRCS-A# msgclean -tx
guest      3  messages (no such user)
3 total qualifying messages.
NRCS-A# msgclean -rx
3 messages removed.
```

When `msgclean -t` is used with the `-a <age in days>` option, information is provided showing which users have messages in a tabular format.

```
NRCS-A# msgclean -t -a 50
loyd      77  messages older than 50 days
jones     3   messages older than 50 days
halls     10  messages older than 50 days
smith     23  messages older than 50 days
...
147 messages older than 50 days.
147 total qualifying messages.
```

Here is an example of how to remove a certain user's messages that are older than the specified age of 50 days:

```
NRCS-A# msgclean -r -a 50 loyd
Do you wish to remove messages older than 50 days to loyd (y/n)? y
77 messages removed.
```


Display or Remove Messages from a Specific User

The `-f <from>` option can be used to display or remove all messages from a specific user. The `<from>` username is case sensitive and literal, meaning wildcards cannot be used. The following example first identifies the number of messages sent from 'monitor' to 'avid', removes them, and then confirms the removal:

```
NRCS-A$ msgclean -tf monitor avid
avid                496 messages from monitor
496 total qualifying messages.
NRCS-A$ msgclean -rf monitor avid
Do you wish to remove messages from monitor to avid (y/n)? y
496 messages removed.
NRCS-A$ msgclean -tf monitor avid
avid                0   messages from monitor
0 total qualifying messages.
```

Wildcard Assumed when No User is Specified

If no username is specified, `msgclean` assumes an asterisk (*) or all users.

```
NRCS-A# msgclean -tx
Jones   8   messages (no such user)
Smith   1   message (no such user)
Taylor  2   messages (no such user)
11 total qualifying messages.
NRCS-A# msgclean -tx jones
Jones   8   messages (no such user)
8 total qualifying messages.
```

offline

```
offline [silent]
```

Puts iNEWS offline. Users cannot log in, but users already on the system can continue normal function. The `silent` option suppresses diagnostics for network connections that are refused because the system is offline. Both `nxserver` and `workserver` typically generate these diagnostics.

The `timestamp` generator, `server timer`, includes an indication that the system is offline in the timestamp messages produced on the console. This provides feedback that the system is offline when the `silent` option was specified.

The `status` command will print “OFFLINE (silent)” when the system is offline with the `silent` option turned on.

You can keep the system offline and turn off the `silent` option by reentering the `offline` command without the `silent` option.

online

```
online
```

Puts iNEWS online. Users can log in and use the system.

otod

```
otod <number> ...
```

otod Command Options

leading 0	Octal
leading 0x	Hex
leading =	Next characters (as characters)
leading -	Negative number
leading _	Two-character compose sequence
leading %	strftime format - subsequent numbers formatted with format for local and gmt times
leading L	locale to use for date formats
leading U =>	following hex byte sequence converted to UTF16 value

This 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
32.0.0.0 Wed Dec 31 16:00:32 1969
```

In this listing, h stands for hexadecimal, o for octal, d for decimal, and u for unsigned decimal. The number conversions are followed by the corresponding ASCII character (space, in this case), and the date value. The 32.0.0.0 is the number shown in IP address format.

The % and L options can assist you in handling different locale settings and formats. The following are two examples:

Type: **otod %X 32**

A message similar to the following appears:

```
h(0x20) o(040) d(32) u(32) SP
0.0.0.32 Wed Dec 31 18:00:32 1969
lcl: 18:00:32
gmt: 00:00:32
```

Type: **otod Lfr 32**

A message, with the date in French, similar to the following appears:

```
locale is fr
h(0x20) o(040) d(32) u(32) SP
0.0.0.32 mercredi 31 décembre 1969, 18:00:32
```

reconnect

```
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). The `reconnect` command must be entered on all of the currently connected servers as well as the server to be reconnected. The `reconnect` command will exit with a diagnostic if the specified server is already connected.

Connects a server to a running iNEWS newsroom computer system. You must type the diskclear command first on the server being reconnected before reconnecting it to the iNEWS newsroom computer system.

remove

```
remove <pathname>
```

Removes specified queues or folders from the database. The queue or folder specified in <pathname> must be empty for the remove command to succeed.

For example:

```
NRCS-A# remove show.6pm.special
```

rename (Superuser only)

```
rename [- u|v|r] <old name> <new name>
```

rename Command Options

- u Update user records only, changing mail, destination, and home directory entries that match <old name>
 - v Verbose output
 - r Resume interrupted operation
-



System must be offline. 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.

reorder

```
reorder <parent> <child> <position>
```

This allows you to change the order of a directory by putting one of the child entries into a new position in the directory. Reordering is only done for Sequential directories.

The <position> parameter can be:

```
-a <sibling> (place <child> after <sibling>)
-b <sibling> (place <child> before <sibling>)
-<number> (move <child> up <number> positions)
+<number> (move <child> down <number> positions)
<number> (place <child> at <number> position)
```

The <child> and <sibling> parameters do not include the full pathname; only the last level name relative to the parent name.

restart

```
restart [-v] <device> | all
```



If you restart a device when a user is editing, data could be lost.

Restarts one or more devices. 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 txnet 41, type: **restart 41**

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.

searchtape

```
searchtape [on <device>] [from <date> to <date>]
[<max # stories>] for <word> ...
```

```
searchtape [on <device>] [just <date>]
[<max # stories>] for <word> ...
```

```
<range> is one of:
    from <date>
    from <date> to <date>
    to <date>
    just <date>
<date> is YY, YYYY, YYYYMMDD or CCYYYYMMDD.
```

Use one of the following date formats: YY, YYMM, YYMMDD, or CCYYMMDD.

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 instant and intersystem messages to users from the console. Messages can be sent to the special user—`computer`—on another system by using the username, `computer@other-system`. These messages are printed on the console of the other system.

Here are two examples of messages sent to user Smith:

```
send smith "are you editing ap-arson"
```

```
send smith "please log out now"
```

sendlong

```
sendlong <user>
```

The long message to be sent is read from standard input—CR LF stripped.

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 (Superuser only)

```
sitedump [-d<device> | -f<file>] [-ehv]
```

Makes backups of your system's site files.

sitedump Command Options

- d <device> is the name of a tape device. The actual device name used will be /dev/<device>. The default is /dev/tape.
 - f <file> is the name of a file to use as the archive.
 - e Eject tape when done.
 - h Help - show usage information.
 - v Verbose
-

siterestore (Superuser only)

```
siterestore [-d<device> | -f<file>] [-ehtv] [<file> ...]
```



All 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, run the `grpcheck` command to rebuild the mail aliases file.

siterestore Command Options

- d <device> is the name of a tape device. The actual device name used will be /dev/<device>. The default is /dev/tape.
- f <file> is the name of a file to use as the archive.
- e Eject tape when done.
- h Help - show usage information.
- t Show table of contents; no files are restored.
- v Verbose

siterestore Command Options

<file> ... Only the named files are restored. If a directory is specified, all files in the directory will be restored.

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.

To list basic system information, type: **status**

A message similar to the following appears:

```
A is online  ID is NRCS
System is AB. Master is A.
Disk status is UNKNOWN.
```

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.

Another example of a status message is:

```
A is online and has been CONFIGURED. ID is NRCS
System is ABC. Master is A.
Disk status is OK. The database is OPEN.
```


If the checkpoint partition is active, the status report will display the last line in the message similar to the following: `The database is OPEN with checkpoint.`

To list system options set in the system profile, type: **status all**

To list the system's license information, type: **status license**

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 Linux command allows you to enter superuser mode. Type su, then type the superuser password when prompted.

The display looks similar to the following:

```
NRCS-A$ su
```

```
password:
```

```
SU: so /dev/console
```

unbusy

```
unbusy [-i][-v][-e][--] <queue name>|<project uuid>
```

Removes edit and order locks from the specified queue in your database.

unbusy Command Options

-i	Ignore the inverted database trait.
-v	Verbose to show the entire story (implies -e).
-e	Show empty fields.
--	Use when <queue name> begins with a hyphen (-).



If a user is actually working in the file, removing the lock could cause data loss.

utraits (Super user only)

```
utraits <username> [<option> <value>] [+|- flag]
```

utraits Command Options

clone	remove (no value)
destination	list (no value)
editmode	realname
keyboard	su
readrate	home
mail	password

The clone option allows creation of new user accounts via cloning an existing account, which will facilitate writing scripts to create large numbers of users in an automated fashion. The remove option allows removal of a single user account.

The list option is intended to be used when the specified name is a CSV or LDIF file.



The bloblist option is useful for removing iNEWS Workstation user preferences should they become corrupted.

utraits Command Flags

b	Blacklist	hr	Highlight read
bloblist	Remove blob list	mp	manage projects
c	Connect	localonly	Local login only
cc	Configure colors	log	Log client/server communication
cs	Configure shortcut toolbar	o	Order
er	Enter and remove	s	Simplified user interface
eu	External user	vb	Video browse
ka	Kill all		

Sets a user's traits. For instance, to assign keyboard 12 to user Jones, type:

```
utraits jones keyboard 12
```

The <username> can be "all", a pattern, or a user name. You can enclose the real name in quotation marks to allow for embedded spaces. Use "<string>*" to indicate wildcards. For instance, "a*" applies to all users with names starting with the letter A.

For importing users from a Windows Active Directory, use list to see what user names are contained in the CSV or LDIF file. They are imported as external users.

- csv:<file> - Use CSV formatted file as list of users to process.
- ldif:<file> - Use LDIF formatted file as list of users to process.

You can modify multiple accounts when importing from a CSV or LDIF file.

version

```
version [--alternate pattern] <filename> ...
```

```
version [+]
```

Displays the version and platform of the iNEWS software you are using.

Type: **version**

A message similar to the following appears and displays product version information:

```
(c) 2000-2009
```

```
This computer program is protected by copyright law and international
treaties. Unauthorized reproduction or distribution of this program, or any
portion of it, may result in severe civil and criminal penalties.
```

```
U.S. GOVERNMENT USERS RESTRICTED RIGHTS: Use, duplication or disclosure by
the U.S. Government is subject to restrictions as set forth in subparagraph
(c) (1) of FAR clause 52.227-19, COMMERCIAL COMPUTER SOFTWARE - RESTRICTED
RIGHTS or, in the case of the Department of Defense or its subcontractor, is
subject to DFARS 227.7202-3, RIGHTS IN COMMERCIAL COMPUTER SOFTWARE OR
COMMERCIAL COMPUTER SOFTWARE DOCUMENTATION.
```

```
ICU License - ICU 1.8.1 and later
```

```
COPYRIGHT AND PERMISSION NOTICE
```

```
Copyright (c) 1995-2003 International Business Machines Corporation and
others
```

```
All rights reserved.
```

```
[...]
```

```
version: 3.0.0.238 RH5
```

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: 3.0.0.238 RH5
```

wholockedit

```
wholockedit [-a][-e][--](<queue name>|<project uuid>)
```

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 option `-a`.

wholockedit Command Options

<code>-a</code>	See all stories.
<code>-e</code>	Show empty fields.
<code>--</code>	Use when <queue name> begins with a hyphen (-).

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 servers and tx links.

blockmode

```
blockmode [yes|no]
```

Only for txnet. Comes after scan line(s) and before an open instruction. This is to allow multiple stories to be transferred over a single TCP data connection instead of a new connection for each story. This uses the FTP mode command with a B parameter.

Some customers send many changing stories via FTP to third part FTP servers. This can consume many TCP port numbers and with a large TIME_WAIT setting prevents the reuse of those port numbers. This can lead to "port exhaustion," that is no ports available for further transfers. When using Stream Transfer Mode, the end-of-file is denoted by the closing of the data connection, hence the need for a new connection for every transfer.

Block Transfer Mode includes end-of-file information in the data so the data connection can be kept open for successive data transfers. This can be used when the iNEWS proprietary FAST mode cannot be used.

If Yes or No is not specified after the blockmode command, the value of Yes is assumed.



To function correctly, the blockmode command must be placed after the scan/bscan/poll/bpoll command and before the open command. The blockmode command will take precedence over the fast command default.

bpoll

```
bpoll <queue name> [<polling interval> [<polling duration>]]
```

Works like poll, except it reads stories in the primary queue in reverse direction. If the polling interval and duration are not specified, a single scan of the primary queue will be done. Applies to timed-interval action servers and tx links.

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 servers and tx links.

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 when it has more than one queue to scan. The everyentry option forces the server to process each entry in a queue, not just modified entries.

charset

```
charset <iconv character set name>
```

Only for txnet. Default character set is UTF-8. Comes after SCAN and applies to all following connections until changed. Can be changed for each connection. Use the Linux utility `iconv --list` to list known coded character sets.

distribution

```
distribution <distribution story queue> [<error queue>]
```

Specifies the queue containing the distribution story and, optionally, an error queue for stories whose distribution codes cannot be processed. Applies to distribution and parallel servers.

dup

```
dup <destination queue> [<distribution code>]
```

Copies the stories in the scan queue to a queue you specify, optionally including a distribution code with them. Applies to action servers and tx links.

every

```
every <dd.hh>
```

Specifies the interval at which a task is performed. You can set this value in days and/or hours. Applies to action servers and tx links.

extension

```
extension <file extension>
```

Applies to tx links. When transmitting stories using HTML format, each transmitted story has a filename composed of a hexadecimal representation of the entry's qstamp and a filename extension of .html. Use the extension command to specify a different extension. Do not include the period in the <file extension> parameter. The defined extension will also be used in place of html for the published file. See the publish job list command for more information.

fast

```
fast [yes | no]
```

Including fast no in a job list (before the open) forces txnet to use separate connections for all data transfers and commands. This command is used if firewalls or WAN accelerators cause interruption in txnet connections. The default is fast yes, which means iNEWS uses the fast FTP protocol and shares a single connection for data transfers and for commands.

ignore

```
ignore [yes | no]
```

Including ignore yes in a job list that performs validation ensures the server accepts any values for the fields it is validating. The default is ignore no. Applies to action servers and tx links.

ignore-del

```
ignore-del
```

Causes a server to take no action when a story is deleted from its scan queue. Applies to action servers and tx links.



This option is not reset for each scan/bscan command set. Use send-del to restore processing of deleted queue entries.

local

```
local <queue name>
```

Specifies the primary wire queue. Applies to parallel servers.

mailto

```
mailto <recipient>...
```

Mails the story as an e-mail text message to each recipient. The list of recipients is a space-separated list. If the sendform option is on, the content of fields in the story is included at the start of the e-mail text message. Only fields present in the story form assigned to the story are included. Each field is identified by the label text associated with the field in the story form assigned to the story.

move

```
move <destination queue> [<distribution code>]
```

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. Applies to action servers and tx links.

number

```
number <form field> <length> <error queue>
```

Assigns a unique number to each story as the story is processed. Specify the form field that will contain the number and the number of digits for the number. Applies to action servers and tx links.

on

```
on <day> ...
```

Indicates on which days of the week a time-interval task will occur. You can combine this instruction with the at keyword to indicate both day and time. Applies to action servers and tx links.

open

```
open <computer> <username> [<format> [<queue name> [<story name>]]]
```

Initiates a network connection to a remote system for story transfer. The `username` you specify must exist with identical passwords on both the local and remote systems. Applies to tx links.

The `<computer>` can include a port number. The format is `host:port`. If no `:port` is included in the `<computer>` parameter, the port defined by the rxnet service is used, if defined; otherwise, the port defined by the FTP service is used.

The `<username>` can specify a simple name or a name in the format `name@host`. Only the name portion is used to locally look up a password. The entire username is sent in the FTP USER command. This may allow connections to be made through proxy servers.

The format, if specified, must be one of the following: 2nsm1, nsm1, or html.

The queue name and story name are only used when the format is set to HTML. The queue name and story name are used to get the Web publishing template that controls the formatting of the story into an HTML page. If the queue name is not specified, the template is taken from the SYSTEM.WEBFORMS queue. The story name can be used to select a specific template from the queue. If not specified, the first story in the queue is used.

order

```
order [yes | no]
```

Indicates that order changes in the 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.

The order command applies to tx links and action servers. For action servers, the order of the scan queue is propagated to each of the dup, move, and replace command destination queues. Specifying order without yes or no is the same as order yes.

passive

```
passive [yes|no]
```

Only for txnet. Comes after scan line(s). Applies to all following connections until changed. Can be changed for each connection. Causes txnet to use passive FTP connections instead of active FTP connections for data transfers.

Data connections are normally initiated by the FTP server when the default active mode is in use. The FTP client (txnet) provides a TCP port number to the FTP server to connect to. Passive mode reverses the sequence. In passive mode, the FTP server provides a TCP port number to the FTP client to connect to. This has significance in how firewalls are setup. If Yes or No is not declared, passive all by itself assumes a value of Yes. In order to function correctly, the passive command must be placed after the scan/bscan/poll/bpoll command and before the open command. The passive command will take precedence over the fast command default.

poll

```
poll <queue name> [<polling interval> [<polling duration>]]
```

Reads stories in the specified primary queue in a forward direction at certain intervals for a specified duration. Polling is used in conjunction with the put or dup commands for rundown mirroring. Stories with modified times greater than the time of the last scan are processed. If an interval and duration are not specified, a single scan of the primary queue will be done. Applies to timed-interval action servers and tx links.

publish

```
publish [no|yes]
```

When placed following a scan or bscan line, the txnet publishing option, publish no, disables appending information to the PUBLISHED.HTML file on the remote system when using HTML export. The default is set to yes. Applies to tx links.

put

```
put [<queue name>]
```

Sends stories over a tx link to a specified queue on the remote system. Applies to tx links.

The <queue name> parameter is optional. When it is not included, the story is put into the default destination queue of the user on the remote system.

quiet

```
quiet [no|yes]
```

Including quiet no in a job list that performs validation sends a message whenever the server successfully validates a story. The default, quiet yes, means a message is sent only when a story fails validation. Applies to action servers and tx links.

remote

```
remote <queue name>
```

Identifies the secondary wire queue. 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 servers and tx links.

replace

```
replace <destination queue name> [<distribution code>]
```

Works like the dup command, except that it updates stories in the destination queue only when they are already present in the destination queue. It does not add new stories to the destination queue. Applies to action servers and tx links.

scan

```
scan <queue name> [priority | all | everyentry]
```

Specifies the queue monitored by this task. The scan line must come before any instructions that manipulate stories in the queue, like dup or move. Ten stories are scanned at a time from each scan queue; adding `priority` to a scan line means all new or modified stories in that queue are scanned at once. The queue identified in the `scan` 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 servers and tx links.

The `priority` option forces the server to interrupt the scan of another queue if the server receives a mailbox notification. The `all` option forces the server to scan the entire queue instead of the limit of 10 stories when there are multiple queues to scan. 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 to process story deletions in the scan queue; this is the default behavior. Use `ignore-del` to have the server take no action when a story is deleted from a scan queue. Applies to action servers and tx links.

sendform

```
sendform
```

Instructs the Tx link to transmit the full form text of each story, rather than just the story's form name. Applies to tx links.

sendpassword

```
sendpassword [yes|no]
```

Only for txnet. Comes after scan line(s) and before any move or remove instructions. Controls the sending of the password for a key locked story.

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. It must include the queue name containing the validation story and an error queue for stories that cannot be validated. Applies to action servers and tx links.

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.

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. If the first character following expect 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. This way, tables and columns you capture retain their format. 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.

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).

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 the string to the screen.

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 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` command. 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 the expected string within the time 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.

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. If you use `wait` without a parameter, the dialog waits until any character is received.

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 iNEWS Workstation.

B System Files

This appendix contains samples of system files you are most likely to reference or change:

- [/etc/hosts](#)
- [/site/config](#)
- [/site/system](#)
- [SYSTEM.CLIENT.WINDOWS](#)
- [SYSTEM.COLORS](#)
- [SYSTEM.CONFIGURE.301-ACTION](#)
- [SYSTEM.GROUPS](#)
- [SYSTEM.INTERPLAY](#)
- [SYSTEM.LISTS](#)
- [SYSTEM.MAP](#)
- [SYSTEM.MOS-MAP](#)
- [SYSTEM.RESOURCE](#)
- [SYSTEM.WIRES.DISTRIBUTION](#)
- [SYSTEM.WIRES.KEYWORDS](#)
- [SYSTEM.WIRES.KEYWORDS-AP](#)
- [SYSTEM.WIRES.KEYWORDS-AP2](#)

/etc/hosts

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          localhost.localdomain localhost

# iNEWS Servers
123.123.123.95    nracs-a1 a1 nracs-a1.local
123.123.123.96    nracs-b1 b1 nracs-b1.local
10.1.1.1          nracs-a  a  nracs-a.local
10.1.1.2          nracs-b  b  nracs-b.local

# FTS
123.123.123.54    ftsserver
```

/site/config

```

; Avid iNEWS Configuration

host      ab      a

          servers 201                ;mail
          servers 211                ;keyword
          servers 231 233 235 237 239 ;seek
          servers 241                ;ftsindex
          servers 251 253 255 257 259 261 263 265 267 269 ;action
          servers 271 273 275 277 279 281 283 285 287 289 ;monitor
          servers 291 293 295 297 299 301 303 305 307 309 ;monitor
          servers 311 313 315 317 319 ;monitor
          servers 321 323 325 327 329 ;txnet
          servers 331 333 335 337 339 341 343 345 347 349 ;rxnet
          servers 351 353 355 357 359 361 363 365 367 369 ;rxnet
          servers 371 373 375 377 379 381 383 385 387 389 ;rxnet
          reslist 391 393 395 397 399 ;console
          reslist 401                ;webaccess
          reslist 501 503 505 507 509 511 513 515 517 519 ;inws
          reslist 521 523 525 527 529 531 533 535 537 539 ;inws
          reslist 601 602                ;wireservers
          reslist 701 703 705 707 709 ;WebClient
          reslist 711 713 715 717 719 ;Instinct
          reslist 801 803 805 807 809 ;Community
          reslist 901 903 905          ;API

;

host      ab      b

```

```
servers 212 ;keyword
servers 232 234 236 238 240 ;seek
servers 242 ;ftsindex
servers 252 254 256 258 260 262 264 266 268 270 ;action
servers 272 274 276 278 280 282 284 286 288 290 ;monitor
servers 292 294 296 298 300 302 304 306 308 310 ;monitor
servers 312 314 316 318 320 ;monitor
servers 322 324 326 328 330 ;txnet
servers 332 334 336 338 340 342 344 346 348 350 ;rxnet
servers 352 354 356 358 360 362 364 366 368 370 ;rxnet
servers 372 374 376 378 380 382 384 386 388 390 ;rxnet
reslist 392 394 396 398 400 ;console
reslist 402 ;webaccess
reslist 502 504 506 508 510 512 514 516 518 520 ;inws
reslist 522 524 526 528 530 532 534 536 538 540 ;inws
reslist 702 704 706 708 710 ;WebClient
reslist 712 714 716 718 720 ;Instinct
reslist 802 804 806 808 810 ;Community
reslist 902 904 ;API
;
host a a
servers 201 ;mail
servers 211 212 ;keyword
servers 231:240 ;seek
;servers 241 242 ;fts
servers 251:270 ;action
```

```
servers 271:320 ;monitor
servers 321:330 ;txnet
servers 331:390 ;rxnet
reslist 391:400 ;console
reslist 401 402 ;webaccess
reslist 501:540 ;inws
reslist 601 602 ;wireservers
reslist 701:710 ;WebClient
reslist 711:720 ;Instinct
reslist 801:810 ;Community
reslist 901:905 ;API
;
host b b
servers 201 ;mail
servers 211 212 ;keyword
servers 231:240 ;seek
servers 241 242 ;fts
servers 251:270 ;action
servers 271:320 ;monitor
servers 321:330 ;txnet
servers 331:390 ;rxnet
reslist 391:400 ;console
reslist 401 402 ;webaccess
reslist 501:540 ;inws
reslist 601 602 ;wireservers
reslist 701:710 ;WebClient
```

```
reslist 711:720 ;Instinct
reslist 801:810 ;Community
reslist 901:905 ;API
```

```
;
```

```
; SERVERS
```

```
;
```

```
;
```

```
;GENERAL
```

```
;
```

```
server 201 mailserver 201 - ;
```

```
;
```

```
; KEYWORD
```

```
;
```

```
server 211 keyword 211 - ;
```

```
server 212 keyword 212 - ;
```

```
;
```

```
; SEEK
```

```
;
```

```
server 231 seek 231 - ;
```

```
server 232 seek 231 - ;
```

```
server 233 seek 231 - ;
```

```
server 234 seek 231 - ;
```

```
server 235 seek 231 - ;
```

```
server 236 seek 231 - ;
```

```
server 237 seek 231 - ;
```



```
server      238      seek      231      -      ;
server      239      seek      231      -      ;
server      240      seek      231      -      ;
;
; FTS
;
;server      241      ftsindex  241      -      ;FTSINDEX
;server      242      ftsseek   242      -      ;FTSSEEK
;
; ACTION
;
server      251      action    251      -      ;Phones
server      252      action    252      -      ;Shredder
server      253      action    253      -      ;Assignment Archive
server      254      action    254      -      ;Auto archive
server      255      action    255      -      ;
server      256      action    256      -      ;
server      257      action    257      -      ;
server      258      action    258      -      ;
server      259      action    259      -      ;
server      260      action    260      -      ;
server      261      action    261      -      ;
server      262      action    262      -      ;
server      263      action    263      -      ;
server      264      action    264      -      ;
server      265      action    265      -      ;
```

```
server      266      action      266      -      ;
server      267      action      267      -      ;
server      268      action      268      -      ;
server      269      action      269      -      ;
server      270      action      270      -      ;

;

; MONITOR

;

server      271      monitor     271      -      ;show.mos.rundown
server      272      monitor     272      -      ;show.training.rundown
server      273      monitor     273      -      ;
server      274      monitor     274      -      ;
server      275      monitor     275      -      ;
server      276      monitor     276      -      ;
server      277      monitor     277      -      ;
server      278      monitor     278      -      ;
server      279      monitor     279      -      ;
server      280      monitor     280      -      ;
server      281      monitor     281      -      ;
server      282      monitor     282      -      ;
server      283      monitor     283      -      ;
server      284      monitor     284      -      ;
server      285      monitor     285      -      ;
server      286      monitor     286      -      ;
server      287      monitor     287      -      ;
server      288      monitor     288      -      ;
```

server	289	monitor	289	-	;
server	290	monitor	290	-	;
server	291	monitor	291	-	;
server	292	monitor	292	-	;
server	293	monitor	293	-	;
server	294	monitor	294	-	;
server	295	monitor	295	-	;
server	296	monitor	296	-	;
server	297	monitor	297	-	;
server	298	monitor	298	-	;
server	299	monitor	299	-	;
server	300	monitor	300	-	;
server	301	monitor	301	-	;
server	302	monitor	302	-	;
server	303	monitor	303	-	;
server	304	monitor	304	-	;
server	305	monitor	305	-	;
server	306	monitor	306	-	;
server	307	monitor	307	-	;
server	308	monitor	308	-	;
server	309	monitor	309	-	;
server	310	monitor	310	-	;
server	311	monitor	311	-	;
server	312	monitor	312	-	;
server	313	monitor	313	-	;
server	314	monitor	314	-	;

```
server      315      monitor      315      -      ;
server      316      monitor      316      -      ;
server      317      monitor      317      -      ;
server      318      monitor      318      -      ;
server      319      monitor      319      -      ;
server      320      monitor      320      -      ;

;

; RESOURCES

; TXNET

;

server      321      txnet      321      -      ;
server      322      txnet      322      -      ;
server      323      txnet      323      -      ;
server      324      txnet      324      -      ;
server      325      txnet      325      -      ;
server      326      txnet      326      -      ;
server      327      txnet      327      -      ;
server      328      txnet      328      -      ;
server      329      txnet      329      -      ;
server      330      txnet      330      -      ;

;

; RXNET

;

server      331      rxnet      -      -      ;
server      332      rxnet      -      -      ;
server      333      rxnet      -      -      ;
```

```
server      334      rxnet    -      -      ;
server      335      rxnet    -      -      ;
server      336      rxnet    -      -      ;
server      337      rxnet    -      -      ;
server      338      rxnet    -      -      ;
server      339      rxnet    -      -      ;
server      340      rxnet    -      -      ;
server      341      rxnet    -      -      ;
server      342      rxnet    -      -      ;
server      343      rxnet    -      -      ;
server      344      rxnet    -      -      ;
server      345      rxnet    -      -      ;
server      346      rxnet    -      -      ;
server      347      rxnet    -      -      ;
server      348      rxnet    -      -      ;
server      349      rxnet    -      -      ;
server      350      rxnet    -      -      ;
server      351      rxnet    -      -      ;
server      352      rxnet    -      -      ;
server      353      rxnet    -      -      ;
server      354      rxnet    -      -      ;
server      355      rxnet    -      -      ;
server      356      rxnet    -      -      ;
server      357      rxnet    -      -      ;
server      358      rxnet    -      -      ;
server      359      rxnet    -      -      ;
```

```
server      360      rxnet    -    -    ;
server      361      rxnet    -    -    ;
server      362      rxnet    -    -    ;
server      363      rxnet    -    -    ;
server      364      rxnet    -    -    ;
server      365      rxnet    -    -    ;
server      366      rxnet    -    -    ;
server      367      rxnet    -    -    ;
server      368      rxnet    -    -    ;
server      369      rxnet    -    -    ;
server      370      rxnet    -    -    ;
server      371      rxnet    -    -    ;
server      372      rxnet    -    -    ;
server      373      rxnet    -    -    ;
server      374      rxnet    -    -    ;
server      375      rxnet    -    -    ;
server      376      rxnet    -    -    ;
server      377      rxnet    -    -    ;
server      378      rxnet    -    -    ;
server      379      rxnet    -    -    ;
server      380      rxnet    -    -    ;
server      381      rxnet    -    -    ;
server      382      rxnet    -    -    ;
server      383      rxnet    -    -    ;
server      384      rxnet    -    -    ;
server      385      rxnet    -    -    ;
```

```
server      386    rxnet    -    -    ;
server      387    rxnet    -    -    ;
server      388    rxnet    -    -    ;
server      389    rxnet    -    -    ;
server      390    rxnet    -    -    ;

;

; NETWORK CONSOLE CONNECTS

;

resource    391    console          -    ;
resource    392    console          -    ;
resource    393    console          -    ;
resource    394    console          -    ;
resource    395    console          -    ;
resource    396    console          -    ;
resource    397    console          -    ;
resource    398    console          -    ;
resource    399    console          -    ;
resource    400    console          -    ;

;

; WEB ACCESS

;

websession  401                                ;
websession  402                                ;

;

; INWS RESOURCES

;
```

```
inws          501:540    -    gnews    -    ;
;
;DATA RECEIVER
wireserver   601        news    AP    -    ;Associated Press
wireserver   602        news    EM    -    ;E-Mail
;
;
;WEB CLIENT RESOURCES
;
webclient    701:710    -    gnews    -    ;
;
;iNEWS INSTINCT RESOURCES
;
aiws         711:720    -    gnews    -    ;
;
;INWS (COM and RSEARCH) SESSIONS
cinws       801:810    -    gnews    -    ; Community
api         901:905    -    gnews    -    ;API
```


/site/system

```
id=NRCS      net=AB

lowwater=5000  highwater=6250    purgelimit=5

load=5        pausetimeout=0:05

;

; defaults - if parameter not present these values are assumed
;           unless overridden on the command line.

;

; auto_upgrade=yes      lowwater=1250      remotetimeout=0:00
; clockmax=12           maxhits=500        security=or
; excluded_video=none   min_passwd_length=5    timechar=:
; highwater=2500        msgserver=silent    timer=verbose
; lastlogin=yes         pausetimeout=0:30    wordlength=0
; load=0                purgelimit=0
; localtimeout=0:00     readrate=180
```

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.COLORS

Further information about this System file is provided in [“SYSTEM.COLORS” on page 209](#).

Example:

```
1          255          000          000          ; red
2          000          255          000          ; green
3          255          255          000          ; yellow
```

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.misc
          dup   phones.*all
```

SYSTEM.GROUPS

Information about this System file/directory is provided in the chapter “Groups” on page 155.

SYSTEM.INTERPLAY

The queue SYSTEM.INTERPLAY (/site/dict/queues: Q_INTERPLAY) must exist for an iNEWS workstation to be able to play low-res streams created by Interplay. The first story in SYSTEM.INTERPLAY is a four-column story.

The first column is the index of the Interplay server, starting with 0. The second column is the hostname of the Interplay server. This name must be resolvable by the iNEWS Workstation. The third column is an optional column holding the username to be used when connecting to Interplay. The fourth column is an optional column holding the password to be used when connecting to Interplay. If the third and fourth columns are blank, the iNEWS username and password will be used to connect to Interplay.

```
0      interplay1
1      interplay2      streamuser      streampass
```



Several Interplay components are required for this workflow to be supported: iNEWS Instinct, Interplay Transcode, Interplay Stream Publish, Interplay Media Services, and Interplay Stream Server. QuickTime 7.5 must also be installed on the iNEWS workstation.

When Instinct associates a sequence with an iNEWS story, the iNEWS workstation will display a context option called “Play Low-res Video.” When this option is selected, iNEWS will contact the Interplay server to retrieve the streaming URL. If the sequence has been mixed down and published to a streaming server, iNEWS will then disconnect from the server and open a QuickTime window inside the iNEWS client. The QuickTime application will automatically play the available stream.

SYSTEM.LISTS

Information about this System file is provided in “SYSTEM.LISTS” on page 209.

SYSTEM.MAP

```

;
;<rundown queue>          <event dir>    <composite> <grp> <mon off time>
;<server type> <server name> <backup server> <CAWS form>
;  video <device name> <alias name> <channel assignment agent>
;  ss <device name> <alias name> <drive>:<directory> <stack name|#>
;  cg <device name> <alias name> <msg drive>:<msg dir> <tmpl drive>:<tmpl
dir><start address> <end address> <address offset>
;  mos <device name> <list secondary device names for redirection>
;
;=====
; Show: Training (using monitor number 272) - COMMAND EXAMPLE
show.training.rundown      show.training      -          monitor      1815
wnasvr  magneto           msn-cmd                command-master
        video      EF                -          E                ;
;        cg        cg                -          1          news        ;
;
;
; Show: Training (using monitor number 320) - CONTROL AIR EXAMPLE
;show.training.rundown    show.training      -          monitor      1815
;casvr   gambit           -          bcs-cg
;        video      video      -          3
;        cg        cg        -          C:NEWS          -          2200      2239 -
;        ss        ss        -          shared:          900
;
;

```

```

; Show: Training (use monitor # 320) - CONTROL AIR EXAMPLE (TWO SERVERS)
;show.mornings.rundown  show.mornings  -          monitor  0710
;casvr  ca-video          -          bcs-video
;      video  video      -          3
;casvr  ca-cg            -          bcs-cg
;      cg      cg        -          C:NEWS  -          2200  2239  20
;
;

```

SYSTEM.MOS-MAP

```

ShowActiveXLaunch = No
ReplaceTime = Yes
TABLE-START DeviceTable
aurora.gvg1.chicago.wavd.mos  aurora1  a b c
aurora.gvg2.dallas.kavd.mos  aurora2  a b c
SONY  SONY
CHYRON  chyron
PILOT.vizrt.chicago.wavd.mos  pilotmos  a b
inscriber.harris.chicago.wavd.mos  harris  A B C
; DEKO section
DEKOMOS  DEKO-MOS 1
DEKO  dekogrp
DekoSelect DEKOID
TABLE-END

```

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	intv1	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	

SYSTEM.RESOURCE

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.medsorce
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

C Standard Dictionaries

This appendix defines and explains contents of standard dictionaries as they are installed on your iNEWS newsroom computer system. Reference this information when modifying dictionary contents. Major sections are:

- [Using Dictionaries to Define Messages and Commands](#)
- [Customizing Dictionaries](#)
- [Utility Messages Dictionary \(/site/dict/messages\)](#)
- [Queues Dictionary \(/site/dict/queues\)](#)
- [Words Dictionary \(/site/dict/words\)](#)
- [Keyboard Macros Dictionary \(/site/dict/keymacros\)](#)
- [Case-shifting Dictionary \(/site/dict/shift\)](#)
- [MCS Dictionary \(/site/dict/mcs\)](#)
- [Job List Command Dictionary \(/site/dict/joblist\)](#)
- [D Messages Dictionary \(/site/dict/dmessages\)](#)
- [S Messages Dictionary \(/site/dict/smessaging\)](#)

Using Dictionaries to Define Messages and Commands

Most commands, messages, and many queues your iNEWS newsroom computer system 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.

The following table lists names and locations of dictionary files.

File Name & Location	Dictionary File Name
/site/dict/messages	Utility Messages
/site/dict/queues	Required Queues
/site/dict/words	Status and Option Words
/site/dict/dmessages	Connect Commands and Messages
/site/dict/smessaging	Connect Commands and Messages
/site/dict/joblist	Job lists
/site/dict/keymacros	Key Names for Macros
/site/dict/shift	Case-Shifting Parameters
/site/dict/mcs	MCS Alarms and Words

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 the following Dictionary Standard Names and Translation table.

Standard Name	Translation
W_ERROR	/ERROR
W_DEST	/destination
W_START	/ON

Each translation is preceded by a slash (/). In commands, the translation represents the minimum a user must enter to execute the command.

The translations may be more than one word long, but the dictionaries have limited space, so keep each translation as short as possible. As a rule, make translations just long enough to be unique from any other translation.

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.



iNEWS newsroom computer system 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 line editor, at the console.

Changing Default Dictionary Values

As an example of how to modify a dictionary translation, change the `enter` directory and `enter queue` commands to make `directory` 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. From the PuTTYCS application, select the PuTTY Filter created for sending server commands to all servers, so changes you make are made to each server's copy of the file.

The `enter directory` and `enter queue` commands have their translations stored in `/site/dict/queues`.



This procedure uses the `ed` line editor.

2. Open `/site/dict/queues` for editing by typing:

```
ed /site/dict/queues
```

A message similar to the following appears:

```
editing /site/dict/queues
1824
```

3. Move to the beginning of the file by typing:

```
1
```

4. Move to the line containing `Q_UNKNOWN` by typing **<ENTER>**.

5. Change `wires.unknown` to `wires.mystery` by typing:

```
s/unknown/mystery
```

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:

```
NRCS-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 s** to see whether anyone is still logged in.

If there is someone logged in, a message similar to the following appears:

```
G500 terry1      A
```

5. If anyone is logged in, type **logout all** to log them out.
6. Type **stop all** to stop all servers.
7. Enter superuser mode at the console.
8. Type **makemontab -i**

After modifying a dictionary, run **makemontab** and **maketab** to have your system read the modified dictionaries and incorporate changes into its programs.

9. Type **maketab -i**

The **maketab** command translates each dictionary and then displays unused space in these dictionaries.

If you changed the name of a command in a command dictionary, you must also change the function key definition that references that command.

10. Restart all devices.
11. At the prompt, exit from superuser using **Ctrl+D**.
12. 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 **makemontab** or **maketab** console command:

```
Translating </site/dict/queues>
```

```
Cannot access translation file </site/dict/queues>.
```

```
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 dictionary default translations without editing the dictionary again. Original dictionary files are stored in the /site/dict directory and have a pound symbol (#) prefix, such as #mcs. So, even after you have made changes to a dictionary, you can restore the standard default translations by copying the appropriate dictionary file from /site/dict and running the makemontab and maketab commands again.

To restore standard translations for Q_UNKNOWN, do the following:

1. List files in /site/dict by running the following ls command:

```
ls /site/dict
```

A display similar to the following appears:

```
dmessages  joblist  keymacros  mcs  messages  queues  shift  shift.CP1250
shift.CP1252  shift.CP1254  shift.CP1256  smessages  words

#dmessages  #joblist  #keymacros  #mcs  #messages  #queues  #shift
shift.CP1251  shift.CP1253  shift.CP1255  shift.CP1257  #smessages
#words
```

2. Type **cp** followed by the pathname of the file you want to copy, and the pathname of the file you want to contain the copy. For instance,

```
cp /site/dict/#queues /site/dict/queues
```

After you copy the file to /site/dict, complete the procedure for changing a translation.

Utility Messages Dictionary (/site/dict/messages)

The /site/dict/messages dictionary holds a number of utility messages displayed in utility programs used by the iNEWS newsroom computer system 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—with some exceptions—the system looks up translations only as they are needed rather than building them into a program. Some exceptions include the following messages, which are built into programs via the maketab program:

- M_COMPUTER
- M_KEYBAD
- M_LASTLOG
- M_ONDEVICE
- M_WIREFAIL
- M_WIREIDLE
- M_WIRERESUME



Some messages defined in dictionary entries on the server that would typically be displayed in the iNEWS client status bar can now have a display method designator to determine if the message should be displayed in the status bar or as a pop-up message. This display method designator is based on the first character of the message definition. A display designator of 1 denotes that the message will appear on the status bar. A display designator of 2 denotes the message displays as a pop-up dialog in addition to being displayed in the status bar and alerts history. The default display method will be a '1' if no display designator is specified.

This section describes these utility messages in the following tables:

- [DBServer Program Messages](#)
- [Disconnect Program Messages](#)
- [Category and Keyword Check Program Messages](#)
- [Keyboard Check Program Messages](#)
- [Keyboard Check Program Messages for Macros](#)
- [Grpcheck Messages](#)
- [Wire Program Messages](#)
- [Mail Server Messages](#)
- [Validation \(Action\) Server](#)
- [Seek Server Messages](#)
- [Last Login Messages](#)
- [Print Server Messages](#)

- [dbtraits Messages](#)
- [Save Error \(Workstation\) Messages](#)

DBServer Program Messages

Standard Name	Translation
M_NOSPACE	/2NO SPACE IN SYSTEM
M_LOWSPACE	/1SYSTEM LOW ON SPACE

Disconnect Program Messages

Standard Name	Translation
M_DISCONNECT	/2COMPUTER DISCONNECTED

Category and Keyword Check Program Messages

Standard Name	Translation
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_BADALWAYS	/Always mismatch on shared destination
M_NOTQUEUE	/Not a queue
M_PURGEZERO	/Queue is never purged
M_SYSERROR	/System error
M_CATLONG	/Category code word too long

Standard Name (Continued)	Translation
M_CATBAD	/Category code word invalid
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_EXPMAX	/Too many expressions
M_WNOTLAST	/Default keyword list must be last
M_SYNEROR	/Syntax error
M_MISSING	/Missing
M_UNEXPCTD	/Unexpected
M_FILENUM	/Maximum file number bad
M_UPDATEQ	/UPDATE queue
M_INVALIDQ	/Invalid destination queue

Keyboard Check Program Messages

Standard Name	Translation
M_KEYDUP	/Duplicate key description
M_KEYRANGE	/Invalid key number
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_KEYREP	/Warning: a key definition contains a repeating function

Standard Name (Continued)	Translation
M_KEYFUNKY	/Warning: badly placed @ exists in key definition line
M_KEYOK	/Keyboard ok
M_KEYBAD	/Keyboard NOT usable
M_COMPUTER	/Computer

Keyboard Check Program 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
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_UNKNOWN	/unknown macro key identifier: %s
M_IGNORING	/ignoring: %s
M_UNEXPECTED	/unexpected: %s
M_TWODEFS	/duplicate macro definition:
M_LONESTATE	/isolated keyboard state:
M_DISTRIBUTE	/%s does not distribute

Standard Name (Continued)	Translation
M_EMPTY	/empty macro
M_NUMKEYNOSHIFT	/Shifted numeric keypad 0 - 9 keys cannot be assigned macros
M_STDHELP	/Warning: "Help" key redefined:
M_STDCELLEDIT	/Warning: "Edit Cell" key redefined:
M_STDFINDNEXT	/Warning: "Find Next" key redefined:
M_STDEXIT	/Warning: "Exit" key redefined:
M_STDCLOSE	/Warning: "Window Close" key redefined:
M_STDREFRESH	/Warning: "Refresh" key redefined:
M_STDTOGGLETEXTFORM	/Warning: "Toggle Story Form" key redefined:
M_STDSCRIPT	/Warning: "Script Swap" key redefined:
M_STDPRIORITYQUEUE	/Warning: "GoTo Priority Queue" key redefined:
M_STDALERTSHISTORY	/Warning: "GoTo Alerts History" key redefined:
M_STDMESSAGEBAR	/Warning: "Communicate Message Bar" key redefined:
M_STDMESSAGEHISTORY	/Warning: "Communicate Message Show History" key redefined:
M_STDMAIL	/Warning: "Communicate Open/Close Mail" key redefined:
M_BADPAUSE	/Invalid pause interval

Grpcheck Messages

Standard Name	Translation
M_GRPBADALIAS	/Invalid name follows word "alias"
M_GRPBADGRP	/Invalid name follows word "group"
M_GRPBADDEV	/Not a workstation device
M_GRPERRSKIP	/Group or alias word missing. Skipping text

Standard Name (Continued)	Translation
M_GRPEXINPUT	/Ignoring words following group name
M_GRPAEXINPUT	/Ignoring words following alias name
M_GRPINT	/Internal groupchecker error
M_GRPDBERR	/Failed to access first story
M_GRPDEVSYN	/Bad workstation device specification
M_GRPNAMUSR	/User name used as group or alias name
M_GRPNAMDUP	/Duplicate group or alias name
M_GRPNONAME	/Missing group name
M_GRPALNONAME	/Missing alias name
M_GRPGRPREC	/Recursive group membership
M_GRPINALIAS	/Name already used as group name
M_GRPAINGRP	/Name already used as alias name
M_GRPMEMBAD	/Not a user or workstation
M_GRPNIX	/No groups or aliases found
M_GRPRESERV	/Improper use of reserved word
M_GRPNOSTORY	/Failed to open story
M_GRPNOQ	/Failed to open queue
M_GRPFLODEF	/Cannot open default aliases file
M_GRPFLONEW	/Cannot open new aliases file
M_GRPFLSOLD	/Cannot save old aliases file
M_GRPFLLONG	/more than 50,000 alias names created
M_GRPPOK	/GROUPS story OK
M_GRPBAD	/GROUPS story NOT OK
M_GRPSSOSO	/GROUPS story accepted, with errors
M_GRPNOSUSER	/Lookup of "so" user in passwd file failed.
M_GRPCHOWNFAILED	/Could not change ownership of mail alias file.

Wire Program Messages

Standard Name	Translation
M_WIREFAIL	/1WIRE COMMUNICATION ERROR
M_WIREIDLE	/1wire has been idle for
M_WIRERESUME	/1wire received story, was idle for

Mail Server Messages

Standard Name	Translation
M_MAILSYNTAX	/Can't send mail, no address
M_MAILNOREC	/Unable to receive mail from
M_MAILQUEUE	/Can't return mail, bad mail queue

Map Check Program Messages

Standard Name	Translation
M_MAPDIRERR	/Failed to open Directory file
M_MAPNOQ	/Failed to open SYSTEM.MAP queue
M_MAPDBERR	/Failed to find story in SYSTEM.MAP queue

Validation (Action) Server

Standard Name	Translation
M_VALID	/Story valid

Standard Name	Translation
M_INVALID	/Story invalid
M_VMOVEDTO	/Story invalid - Moved to

Seek Server Messages



Seek Server Messages are also used by ftsseek.

Standard Name	Translation
M_BGSNSCHP	/No search path
M_BGSNRESP	/No results path
M_BGSSCHTP	/Invalid Search Type
M_BGSIKWEX	/Invalid keyword expression
M_BGSSPDNE	/Search Path does not exist
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_BGSSPI	/Invalid search path
M_BGSRQI	/Invalid results queue
M_BGSRQOE	/Open error on results queue
M_BGSDONE	/Done
M_BGSSTOPPED	/Stopped
M_BGSCANCELLED	/Cancelled
M_BGSSYNTAXERROR	/Syntax Error

Standard Name (Continued)	Translation
M_BGSCOMMERROR	/Communication Error
M_BGSMPAT	/Missing search pattern

Last Login Messages

Standard Name	Translation
M_LASTLOG	/Last Login
M_ONDEVICE	/on device

Print Server Messages

Standard Name	Translation
M_PRINT_BUSY	/Printer is OFFLINE

dbtraits Messages

Standard Name	Translation
M_CHANGE_GROUP	/1Group, form, order, or sortfield changed.

Save Error (Workstation) Messages

Standard Name	Translation
M_SAVE_ERROR	/Error: Story saved in
M_NO_SAVE	/Error: Story could not be saved.

Queues Dictionary (/site/dict/queues)

The Queues dictionary contains names for system queues such as SYSTEM.KEYBOARDS and the Dead queue.

Queues in this dictionary are used by functions within Media Browse. 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 the following Queues Dictionary table. The 8-bit codes can be defined using 7-bit sequences.

Standard Name	Translation
Q_ACCT	/system.account
Q_CATWORDS	/system.wires.distribution
Q_CLIENT_API	/system.client.api
Q_CLIENT_API_VERSIONS	/system.client.api-versions
Q_CLIENT_COM	/system.client.com
Q_CLIENT_COMMUNITY_VERSIONS	/system.client.community-versions
Q_CLIENT_INSTINCT	/system.client.instinct
Q_CLIENT_INTERPLAY_INSTINCT	/system.client.interplay-instinct
Q_CLIENT_MOBILE_IDS	/system.client.mobile-ids
Q_CLIENT_MOBILE_VERSIONS	/system.client.mobile-versions
Q_CLIENT_VERSIONS	/system.client.versions
Q_CLIENT_WEB	/system.client.web
Q_CLIENT_WEB_VERSIONS	/system.client.web-versions
Q_CLIENT_WINDOWS	/system.client.windows
Q_CLIENT_WIRE_SERVER	/system.client.wire-server
Q_COLORS	/system.colors
Q_CONFIGURE	/system.configure
Q_DEAD	/dead

Standard Name (Continued)	Translation
Q_DESTINATION	/notes
Q_FLASH	/wires.advisory.priority
Q_FORMS	/system.forms
Q_FTS	/system.fts
Q_GROUPS	/system.groups
Q_HELP	/system.help.terminal
Q_HOME	/notes
Q_INDEX	/system.index
Q_INTERPLAY	/system.interplay
Q_KEYBOARDS	/system.keyboards
Q_KEYWORDS	/system.wires.keywords
Q_LISTS	/system.lists
Q_MAIL	/mail
Q_MAILERROR	/system.mail.error
Q_MAILOUT	/system.mail.out
Q_MESSAGE	/system.message
Q_MMAP	/system.map
Q_MOS_MAP	/system.mos-map
Q_MRESOURCE	/system.resource
Q_NODEST	/system.unknown
Q_POST_TO_WEB	/system.post-to-web
Q_PROPERTIES	/system.properties
Q_SCRIPT	/system.dialogs
Q_SCRIPT_TEMPLATES	/system.script-templates
Q_SEARCHTAPE	/system.searchtape
Q_SEEK	/system.seek

Standard Name (Continued)	Translation
Q_SERVICE	/system.service
Q_SPELL	/system.spell
Q_STYLES	/system.styles
Q_TITLE_ENTRY	/system.title-entry
Q_UNKNOWN	/wires.unknown
Q_USERROOT	/people
Q_WEBACC_FORMS	/system.webforms
Q_WEBPUB_FORMS	/system.webforms

Words Dictionary (/site/dict/words)

The Words dictionary 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.

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.

Standard Name	Translation
Wire Priorities and Options	
W_FLASH	/IFLASH
W_BULLETIN	/1BULLETIN
W_URGENT	/1URGENT
W_SILENT	/SILENT
W_ALWAYS	/A
W_TRANSMIT	/TRANSMIT
W_WIRE_FORM	/wires
Status Types	

Standard Name (Continued)	Translation
W_MAIL	/mail
W_HOLD	/HOLD
W_LOCKED	/LOCKED
W_READY	/READY
W_NOTREADY	/?
W_NEW	/NEW
W_WIRE	/WIRE
Special Words for Find	
W_AND	/and
W_NOT	/not
W_ANDNOT	/andnot
W_OR	/or
W_ALL	/all
Print Command Options	
W_STORY	/story
W_SCRIPT	/script
W_RUNDOWN	/rundown
W_DIRECTORY	/directory
W_ON	/on
W_PRINT	/print
W_DAYS	/SunMonTueWedThuFriSat
<p>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.</p>	
W_PAGE	/page
Words Relating to the Seek Server	
W_FAST	/fast

Standard Name (Continued)	Translation
W_INDEXED	/INDEXED
W_ACTIVE	/ACTIVE
W_DONE	/DONE
W_ERROR	/ERROR
W_ABORT	/ABORTED
W_DELIMITERS	.,'
W_PENDING	/PENDING
W_SLOW	/slow
Words relating to FTS Server	
W_BINDFTSI	/ftsserver:6100
W_BINDFTSS	ftsserver:6101
Words relating to Projects	
W_PROJECT_ALL	/all
W_PROJECT_QUERY	/query
W_PROJECT_BUCKET	/bucket
Miscellaneous Options	
W_FIRSTDAYOFWEEK	/0
W_DEFAULT_FORM	/default_form
W_WEBACC_FORM	/access_form
W_WEBPUB_FORM	/publish_form
W_LOGTYPES	/C
W_DEST	/destination
W_RESULTS_FORM	/search-results
W_START	/ON
W_OFF	/OFF
W_YES	/yes

Standard Name (Continued)	Translation
W_NO	/no
W_GROUP	/group
W_ALIAS	/alias
W_ANYSTR	/-
W_BLANKSTR	/+
W_RESTRICTED	/restricted (only used by gtraits program and identifies the name used for the restricted user group)
W_NEAR	/near
W_RUNS	/runs
W_STARTMEDIA	/Primary
W_CUE_REF	/¤

Keyboard Macros Dictionary (/site/dict/keymacros)

The keyboard macros dictionary contains names of keyboard keys for use in keyboard macro definitions for the iNEWS newsroom computer system.

Like other dictionaries, the standard name is in uppercase and must not be changed. Translations can be in lowercase, uppercase, or mixed case.

Standard Name	Translation
K_NULL	/null
K_F1	/f1
K_F2	/f2
K_F3	/f3
K_F4	/f4
K_F5	/f5
K_F6	/f6
K_F7	/f7

Standard Name (Continued)	Translation
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
K_DELETE	/delete
K_END	/end
K_UP	/up
K_DOWN	/down
K_LEFT	/left
K_RIGHT	/right
K_SHIFT	/shift

Standard Name (Continued)	Translation
K_CTRL	/ctrl
K_ALT	/alt
K_TAB	/tab
K_ESC	/esc
K_BACKSPACE	/backspace
K_ENTER	/enter
K_PAUSE	/pause
K_REPEAT	/repeat
K_SPACE	/space
K_WINDOW	/window



The `K_WINDOW` token replaces the former syntax [`<dialog name>`] used in macros to define a dialog wait. If a macro should wait for the Local Printing dialog box before continuing, the syntax is: `{window Local Printing}`. The text following the word window must match exactly the title of the dialog box as it appears in the title bar. If the text does not match, the user must press the Escape key to exit the macro.

Case-shifting Dictionary (/site/dict/shift)

This is only used when restoring pre-Unicode systems. The Case-shifting dictionary maps lowercase characters to their uppercase counterparts and vice versa. Media Browse 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 iNEWS newsroom computer system, 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:

```
; 1252 Windows Latin 1 (ANSI)
;
tolower
  65 -  90 ->  97 - 122      ; A - Z -> a - z
138          -> 154
140          -> 156
159          -> 255
192 - 214 -> 224 - 246
216 - 222 -> 248 - 254
end

toupper
  97 - 122 ->  65 -  90      ; a - z -> A - Z
154          -> 138
156          -> 140
224 - 246 -> 192 - 214
248 - 254 -> 216 - 222
255          -> 159
end
```

The character-conversion table the system uses depends on the interface you are using.

- If you are using the DOS, the character-conversion table is based on the DEC Multinational Character Set (MCS).
- If you are using the Media Browse 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.

When editing 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 for use by the iNEWS newsroom computer system. For more information, call Avid.

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.

MCS Dictionary (/site/dict/mcs)

The MCS dictionary contains the following:

- [Device Types Used by Monitor Servers and Drivers](#)
- [Special Strings Recognized by the Monitor Server](#)
- [Error Messages for the Monitor Server](#)
- [Status Reported in Device Status Field](#)

The standard name is in uppercase and must not be changed. The translation can be in lowercase, uppercase, or mixed case.

Device Types Used by Monitor Servers and Drivers

Standard Name	Translation
A_VIDEO	/VIDEO
A_CASVR	/CASVR

Standard Name	Translation
A_CG	/CG
A_MOS	/MOS
A_MOSSVR	/MOSSVR
A_SS	/SS
A_USRDEF	/USRDEF
A_WNASVR	/WNASVR

Special Strings Recognized by the Monitor Server

Standard Name	Translation
A_CHANNELOPEN	/[
A_CHANNELCLOSE	/]
A_CMDDEL	/*
A_COMDEL	/;
A_FLDDEL	/
A_LOCALHYPHEN	/-
A_PLAYLISTREF	/EMBED:
A_VIDEOIDREQ	/ASSIGNID

Error Messages for the Monitor Server

Standard Name	Translation
A_MBINVL	/Invalid mailbox
A_NOTINMAP	/Queue not in map
A_XMAXSTORY	/Exceeded max # stories, monitor exiting
A_MAPOPENERR	/Map story open error
A_RESOPENERR	/Resource story open error
A_NOSERVER	/Server cannot execute

Standard Name (Continued)	Translation
A_NOSTYLELIST	/Unable to obtain CD stylist
A_ENDSVRS	/ENDSVRS
A_BADMAPCG	/Bad Map line: CG device requires range
A_BADMAPSFORM	/Bad Map line:Invalid ControlAir form
A_BADMAPARG	/Bad Map line: Invalid arg count
A_BADMAPSTYPE	/Bad Map line:Invalid server type
A_BADMAPQTME	/Bad Map line: Invalid time value
A_BADMAPDUPE	/Bad Map line: Duplicate device
A_BADMAPDRV	/Bad Map line: Invalid drive specification
A_BADMAPVCHAN	/Bad Map line: Invalid video channel assignment policy
A_BADMAPBADMOS	/Bad Map line: Unrecognized MOS parameter
A_BADMAPNOMOSID	/Bad Map line: No MOSID in MOS-MAP
A_BADMAPCHOICE	/Bad Map line: Choice server-type mismatch
A_BADMAPMIXED	/Bad Map line: Invalid mixed setting
A_BADMAPVALIDATE	/Bad Map line: Invalid validate style setting
A_BADMOSSIZE	/Bad Map line: MOS item too large
A_BADMAPBADSVR	/Bad Map line: device-server type mismatch
A_BADMOSITEM	/Invalid MOS item
A_BADDUPENTRY	/Bad Resource line: Duplicate style entry
A_BADCGLINES	/Bad Resource line: Invalid # CG fields
A_BADRESARG	/Bad Resource line: Invalid arg count
A_BADCGTMPL	/Bad Resource line: Invalid CG template
A_BADSSADDR	/Bad Resource line:Invalid SS address
A_RESERRORS	/Following errors found in resource story
A_INVLDEV	/Invalid device specification
A_INVLSTYLE	/Invalid style

Standard Name (Continued)	Translation
A_INVLADD	/Invalid address
A_INVLARG	/Missing argument
A_XMAXADD	/Exceeded address range
A_NOFORM	/Cannot access form
A_SRTERR	/Error sorting queue
A_DELERR	/Cannot delete from
A_DIRERR	/Directory open error
A_QUEERR	/Queue open error
A_QUEAPPERR	/Queue append error
A_STRYERR	/Story open error
A_STRYCRT	/Story create error
A_STRYWRT	/Story write error
A_SRVNOTINMAP	/Server not in Map
A_SVRNOCOMM	/Network connection failed
A_DEVBUSY	/Refused connection request
A_NOSSADDR	/No SS address specified
A_MULSSADDR	/Multiple SS addresses specified
A_MONEXIT	/Monitor exiting
A_QUITTIME	/Quite Time reached
A_MUSTUNLOAD	/Must first unload
A_CMDSVRLOADED	/Already loaded
A_READY	/OK
A_NOMEM	/Out of memory:
A_VERSMISMATCH	/CA Server version mismatch: story limits
A_DUPSTAMP	/Persistent duplicate queue stamp

Status Reported in Device Status Field

Standard Name	Translation
A_EVERR	/5ERROR
A_EVSTOP	/DONE
A_EVSTDBY	/2STANDBY
A_EVCUING	/2CUEING
A_EVCUED	/2CUED
A_EVNOTRDY	/4OFFLINE
A_EVREADY	/ONLINE
A_EVPLAY	/3PLAY
A_EVPAUSE	/3PAUSED
A_EVREW	/REWIND
A_EVINCMLPT	/TRANSFER

Job List Command Dictionary (/site/dict/joblist)

Job list commands are used for txnet/action job list processing. The first string on each line is a key; do not change it.

Standard Name	Translation
all	/all
at	/at
blockmode	/blockmode
bpoll	/bpoll
bscan	/bscan
charset	/charset
duplicate	/du

Standard Name	Translation
eof	/eof
every	/every
everyentry	/everyentry
extension	/extension
fast	/fast
ignore	/ignore
ignore-del	/ignore-del
mailto	/mailto
move	/mov
number	/number
on	/on
open	/open
order	/or
passive	/passive
poll	/poll
priority	/priority
publish	/publish
put	/put
quiet	/quiet
remove	/rem
replace	/rep
scan	/scan
sendform	/sendform
sendpassword	/sendpassword
send-del	/send-del
validate	/validate

Standard Name	Translation
verify	/verify

D Messages Dictionary (/site/dict/dmessages)

The D message strings apply to the snews and nxserver programs associated with Connect to Service.

Standard Name	Translation
D_AUTHORIZED	/Not allowed
D_BADARG	/Bad argument
D_BADDEST	/Bad destination
D_ERROR	/System error
D_NOARG	/Needs argument
D_OFFLINE	/Offline
D_UNKNOWN	/Unknown command

S Messages Dictionary (/site/dict/smessaging)

The S message strings apply to the snews and nxserver programs associated with Connect to Service.

Standard Name	Translation
S_CCAPTURE	/capture
S_CQUIT	/quit
S_CPAUSE	/pause
S_CSTOP	/stop
S_CHELP	/help
S_CESCAPE	/escape
S_CECHO	/echo

Standard Name (Continued)	Translation
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_SNPROMPT	/cmd>
S_SNCAPON	/Capturing session to
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 save 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

Standard Name (Continued)	Translation
S_REJECT	/REJECT
S_FINISH	/FINISH
S_RXNET	/RXNET
S_CAPHLP	/ <i><queue></i> Capture to queue, or continue after a pause
S_QUITHLP	/End this connect session
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><c></i> Change the escape character to specified character
S_ECHOHLP	/Toggle local character echo
S_HEOLHLP	/Toggle Hard-End-Of-Line on captured data
S_NXNONAME	/Computer not named
S_NXNOCONF	/System not configured
S_NXNOPTY	/No ptys available
S_NXNODEV	/No device available

D Environment Variables

Some features in the Avid iNEWS newsroom computer system 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-based operating systems, and the Registry Editor program.

This appendix includes the following sections:

- [Registry Editor](#)
- [Environment Variables \(Registry Values\)](#)
- [Environmental Variables for Servers](#)

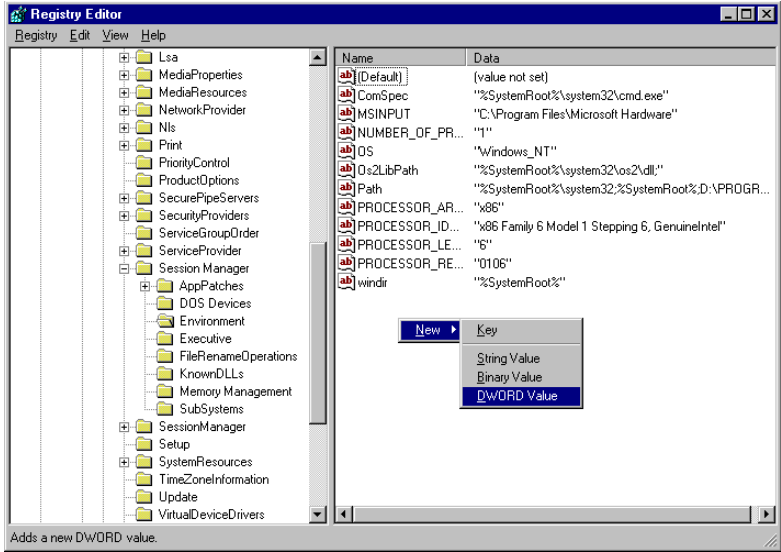
Registry Editor

The Registry Editor is used to create and define environment variables (Registry values) at each workstation. Environment Variables

To access the Registry Editor:


1. Click the Start button on the Windows Taskbar.
2. Select the Run option.
3. Type regedit in the dialog box that appears.

The Registry Editor window appears.



All iNEWS newsroom computer system environment variables are set up and stored in the same location on each workstation. After opening 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-based 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) must be used*

Environment Variables (Registry Values)

Environment variables or registry values are sometimes required to set up certain iNEWS features at various iNEWS Workstations. Environment variables are located and defined in the Registry of iNEWS Workstations—that is, Windows-based PCs running the Client software. The following variables are covered in this section:

- [CCColor](#)
- [DestinationOrder](#)

- [MailLookup](#)
- [MsgMailAlert](#)
- [PIColor](#)
- [ShowTimingBar](#)
- [SyncToServer](#)
- [VT Compatibility](#)

The following sections explain how to can set up environment variables by editing the Registry using the Registry Editor.



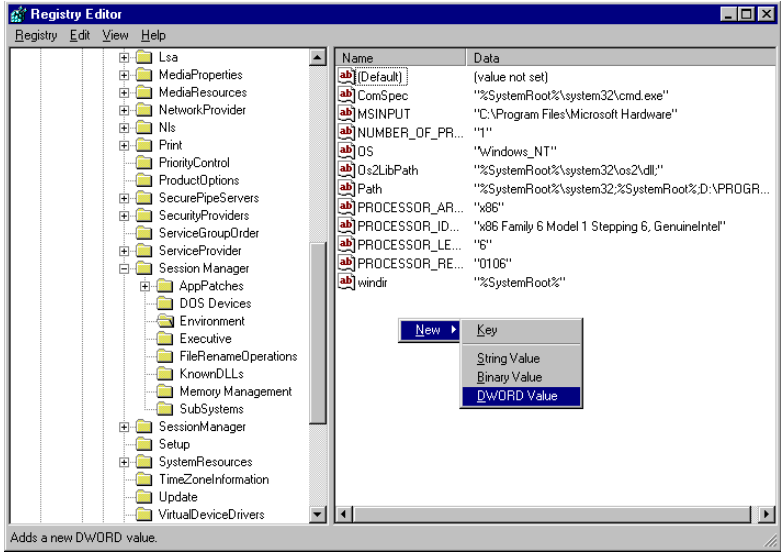
Self-importing files, called reg files, can be executed to automatically import environment variable information into the Registry. These files with their exported registry keys, can be used on PCs running Windows NT -based operating systems. For more information on how to obtain and use these reg files, contact Avid Customer Support.

CCColor

An individual workstation can 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.

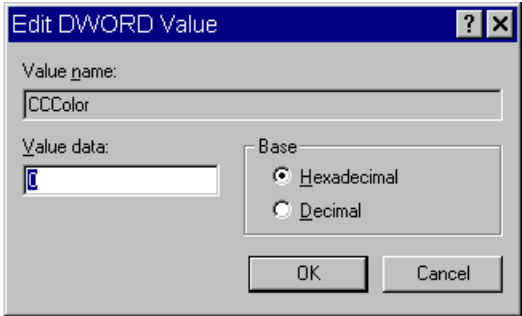
To change the closed captioning text color:

1. Open Registry Editor. See [“Registry Editor” on page 239](#) for more information.
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 appears.



- 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: CCColor.
- 6. Press Enter.
- 7. To set the Value data option:
 - a. Right-click on the CCColor value.
 - b. Select Modify.

The Edit DWORD Value dialog box appears.



- c. Set the Value data using the following 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.
8. Close the Registry Editor window.
 - See “[RGB Hexadecimal Color Chart](#)” on [page 247](#) for more information on possible colors used in this environment variable

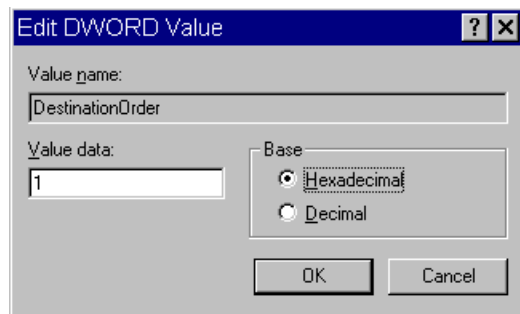
DestinationOrder

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. See “[Registry Editor](#)” on [page 239](#) for more information.
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 appears.
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. Press Enter.
7. To set the Value data option:
 - a. Right-click on the DestinationOrder value.
 - b. Select Modify.

The Edit DWORD Value dialog appears.



- 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.
8. Close the Registry Editor window.

MailLookup

The iNEWS newsroom computer system 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.

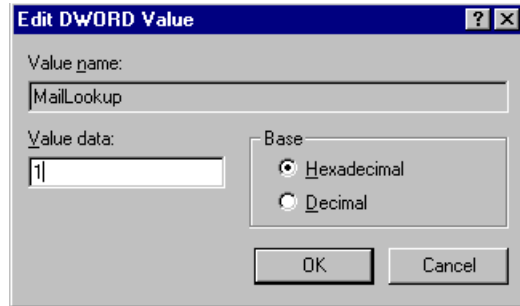


This environment variable must be created and defined in the Registry Editor at each workstation. Default behavior is used at workstations where the environment variable is not defined.

To set the environment variable and hide groups from e-mail lists:

1. Open Registry Editor. See “Registry Editor” on page 239 for more information.
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 appears.
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. To set the Value data option:
 - a. Right-click on the MailLookup value.
 - b. Select Modify.

The Edit DWORD Value dialog appears.



- c. Set the Value data, by typing one of the following 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 MailLookup 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

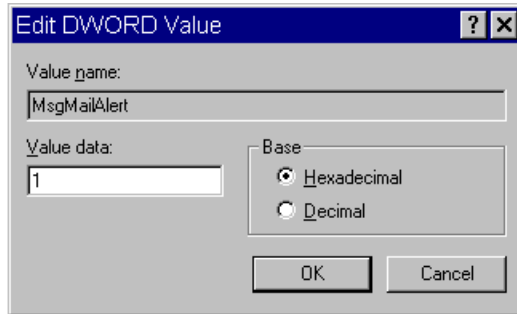
Enabling the Message Mail Alert feature allows you to change the alert behavior so that the iNEWS Workstation will flash message and/or mail alerts on the status bar for only 15 seconds, rather than persistently. By adding the MsgMailAlert variable, you can specify additional settings.

To set the environment variable thereby enabling message mail alerts:

1. Open Registry Editor. See [“Registry Editor” on page 239](#) for more information.
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 appears.
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. To set the Value data option:
 - a. Right-click on the MsgMailAlert value.
 - b. Select Modify.

The Edit DWORD Value dialog appears.



- c. Set the Value data, by typing one of the following options:
 - ▶ 0 - (zero) disable - no alerts whatsoever on status bar
 - ▶ 1 - (one) 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 (one).

- d. Click OK to save the setting and close the dialog box.
8. Close the Registry Editor window.

PIColor

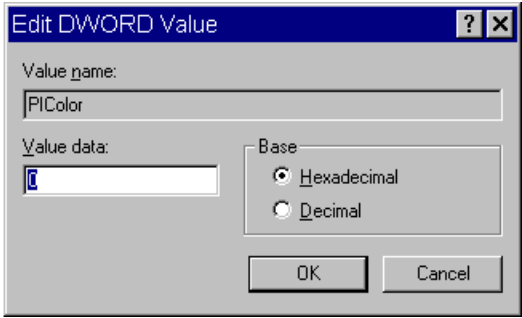
An individual iNEWS Workstation can have its presenter instructions text color changed via an environment variable called PIColor. If no environment variable exists, then the default color of red is used.

To change presenter instructions text color:

1. Open Registry Editor. See [“Registry Editor” on page 239](#) for more information.
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 appears.

- 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. Press Enter.
- 7. To set the Value data option:
 - a. Right-click on the PIColor value.
 - b. Select Modify.

The Edit DWORD Value dialog box appears.



- c. Set the Value data using the following 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 is black because zero corresponds to the color Black.

- d. Click OK to save the setting and close the dialog box.
8. Close the Registry Editor window.

RGB Hexadecimal Color Chart

Avid’s PIColor and CCColor environment variables require RGB Hexadecimal Color codes. Complete RGB Hexadecimal Color Charts, with various color shades, can be found on the Internet, but the following table contains some basic colors, along with their corresponding hexadecimal code values:

Color	Hex	Default
Black	000000	
Blue	0000FF	

Color	Hex	Default (Continued)
Brown	330000	
Green	008800	(Default color for CCColor)
Orange	FF6600	
Pink	CC0099	
Purple	660099	
Red	FF0000	(Default color for PIColor)
White	FFFFFF	
Yellow	FFFF00	

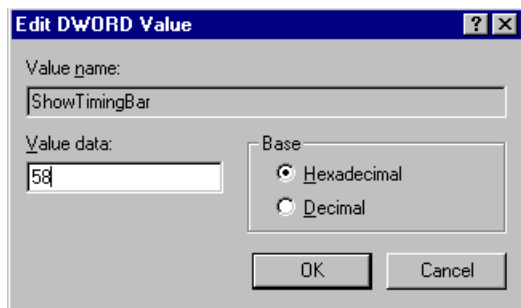
ShowTimingBar

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:

1. Open Registry Editor. See [“Registry Editor” on page 239](#) for more information.
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 appears.
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. Press Enter.
7. To set the Value data option:
 - a. Right-click on the ShowTimingBar value.
 - b. Select Modify.

The Edit DWORD Value dialog box appears.



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 following Scan Codes table for more information.

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

Key	Decimal	Hexadecimal
W	17	11
E	18	12
R	19	13
T	20	14
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

Key	Decimal	Hexadecimal
X	45	2D
C	46	2E
V	47	2F
B	48	30
N	49	31
M	50	32
, < (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

Key	Decimal	Hexadecimal
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
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

- c. Click OK to save the setting and close the dialog box.
8. Close the Registry Editor window.

SyncToServer

The timing feature of iNEWS 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.

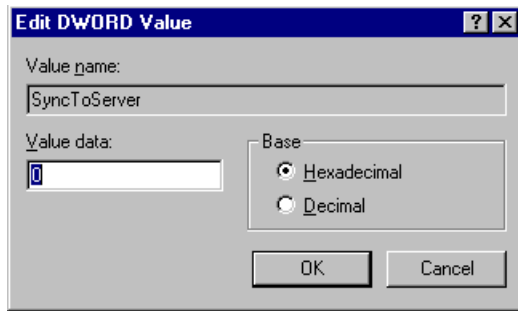


The synchronized timing feature should be enabled only at those workstations used to time a show. If the SyncToServer registry value is not created and defined at a workstation, then the synchronized timing feature is disabled at that workstation.

To enable the synchronized timing feature:

1. Open Registry Editor. See “Registry Editor” on page 239 for more information.
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 appears.
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. Press Enter.
7. To set the Value data option:
 - a. Right-click on the SyncToServer value.
 - b. Select Modify.

The Edit DWORD Value dialog box appears.



- c. Set the Value data by typing one of the following options:
 - ▶ 0 - (zero) disable the synchronized timing feature
 - ▶ 1 - (one) enable the synchronized timing feature
 - d. Click OK to save the setting and close the dialog box.
8. Close the Registry Editor window.

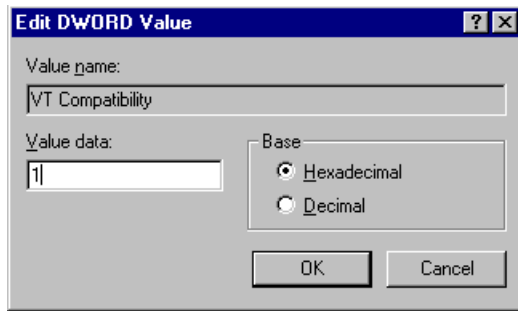
VT Compatibility

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:

1. Open Registry Editor. See “Registry Editor” on page 239 for more information.
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 appears.
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. To set the Value data option:
 - a. Right-click on the VT Compatibility value.
 - b. Select Modify.

The Edit DWORD Value dialog box appears.



- c. Type 1 (one) 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.

DisableCommandLine

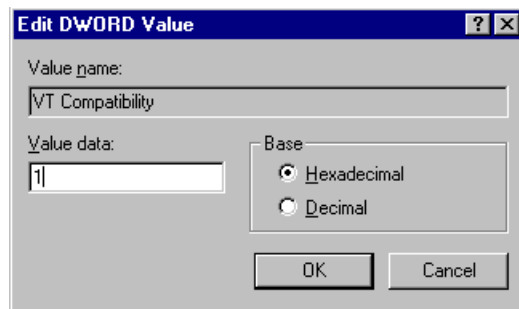
The environment variable called `DisableCommandLine` lets you disable the Command Line option in the Customize Toolbar Button dialog box, thereby restricting users from running executable programs within iNEWS on the PC. `DisableCommandLine` also disables any existing toolbar buttons that utilize the Command Line option.

Environment variables are workstation specific, not user specific; therefore an Environment key must be added to the Registry on any PC on which you want to prevent users from creating custom toolbar buttons that execute programs via the Command Line option. If a registry value is not found in the Environment key of the Registry, then the default behavior—no restriction—is observed at the workstation.

To create and define this value thus disabling the command line option:

1. Open Registry Editor. See “[Registry Editor](#)” on page 239 for more information.
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 appears.
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: `DisableCommandLine`.
6. Press Enter.
7. To set the Value data option:
 - a. Right-click on the `DisableCommandLine` value.
 - b. Select Modify.

The Edit DWORD Value dialog box appears.



- c. Type 1 (one) in the Value data field to disable the Command Line option. If set to zero (0), the Command Line option will remain enabled, so it will not appear grayed out in the Customize Toolbar Button at the iNEWS Workstation.

- d. Click OK to save the setting and close the dialog box.
8. Close the Registry Editor window.

Environmental Variables for Servers

You can control some of the behavior of rxnet by putting some environment variable settings in the `/site/env/rxnet` file.



If the `/site/env/rxnet` file or its parent directory `/site/env` do not exist you need to create the directory using the `mkdir /site/env` command, before you can edit the file. Contact Avid if you need assistance with this procedure.

The specific variables are:

```
RXDEBUG=<level>
```

An `rxdebug` level of 2 produces the command traffic to/from rxnet.

```
RXDEBUGFILE=<filename>
```

The `RXDEBUGFILE` file lets you cause rxnet diagnostics to be put into `<filename>.<process id>` instead of the console.

```
RX_HOT_TO_GO=0
```

When `RX_HOT_TO_GO=0` is set in `/site/env/rxnet`, then rxnet will not print Hot-To-Go diagnostics.

```
RXNOFAST=1
```

The `fast [yes|no]` joblist command allows the fast protocol to be turned off. By default, `fast yes` is used. By setting `fast` to `no`, a user can force txnet to use a separate socket connection for all data transfers. Normally, when txnet connects to an rxnet, they agree to use the command socket connection for data transfers in addition to command exchanges. This option can help resolve problems when connections have a router or firewall between them.

An `rxnofast=1` (one) causes rxnet to refuse attempts to use a fast connection for data transfers. This can be set in the `/site/env/rxnet` file.

```
RXNOPRESENTER=<0|1>
```

Rxnet checks contents of `PRESENTER` field on incoming stories. The story readrate will be

changed if the user name is found. The readrate will be set to the system default if the PRESENTER field is empty. If the PRESENTER name is not found in the user account file, the readrate set in the story is retained. If the environment variable RXNOPRESENTER is present in /site/env/rxnet and is non-zero the old rxnet behavior is used, that is, no readrate adjustment via the PRESENTER field.

```
RXSITEFORMAT=<format>
```

The format controls which transfer protocol is initially used. This can be set to 3.1NSML, 3NSML, 2NSML, or NSML. The default is NSML.



All RXSITE... environment variables are a way to simulate the equivalent “<option>=” SITE commands, intended for use with FTP clients that cannot send those SITE commands. For instance, RXSITEFORMAT enables rxnet support for the “FORMAT=” SITE command. Any SITE command received from the FTP client will override their environment variable counterpart.

```
RXSITECHARSET=<character set name>
```

Rxnet supports the “CHARSET=” SITE command and honors the RXSITECHARSET environment variable setting for the client character set. The list of valid character set names can be produced by using the **iconv-list** Linux console command.

All data received from the client will be converted from the specified character set into Unicode. The sequence “U+HHHH” will be converted into a Unicode value where “HHHH” is the hexadecimal value.

```
RXSITELISTSZ=<size>
```

Rxnet supports the “LISTSZ=” SITE command. The list size controls the number of stories included in a queue list when rxnet responds to an FTP list command. The default value for this is 300.

```
RXSITESENDFORM=<0|1>
```

Rxnet supports the “SENDFORM=” SITE command and honors the RXSITESENDFORM environment variable setting. A value of 1 (one) means to include embedded forms when exporting a story to a client. A value of 0 (zero) means to omit embedded forms. This is the default and the current behavior. This feature allows form stories to be exported.

```
RXSITEIDLE=<seconds>
```

The idle time value may be set from the client via `SITE IDLE=<seconds>` where *<seconds>* is a count of seconds. For instance:

```
SITE IDLE=900 sets the idle timeout to 15 minutes
```

```
SITE IDLE=0 disables the timeout
```

The client can query the rxnet server to find out what the current idle timeout is with `SITE IDLE`. You can also define an idle timeout value to override the `localtimeout` setting by inserting `RXSITEIDLE=<seconds>` in the file `/site/env/rxnet`.

```
RXOLDNLST=1
```

FTP clients that use the `MGET` command make an `NLST` request to get the list of story/file names for retrieval. The `LIST` and `NLST` commands produce the identical output, a directory listing. The `NLST` command produces a name list, that is a list of subdirectories in the directory and a list of stories id's in a queue. The iNEWS FTP Server produces a list of names without story attributes. If you want the iNEWS Server to include story attributes, put `RXOLDNLST=1` in the rxnet environment. If you do not want story attributes included, use `RXOLDNLST=0`.

E Managing Traits at the Console

Chapter 4 in the *iNEWS Newsroom Computer System Setup and Configuration Manual* explains how the system administrator can access and change the various user traits associated with each user's account from an iNEWS Workstation. In that same book, Chapter 5 explains how to manage database traits from a workstation, and Chapter 6 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 console.

This appendix contains the following main sections:

- [Viewing User Traits from the Console](#)
- [Modifying User Traits from the Console](#)
- [User Traits Console Command Summary](#)
- [Changing Database Traits from the Console](#)
- [Database Traits Console Command Summary](#)
- [Managing Group Traits at the Console](#)

Viewing User Traits from the Console

From the 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:

```
NRCS-A$ list u-v danielmi
```

The verbose result of the command will look something like this:

```

user      rr  kb  su  m  l  SOEKCVHP  sc  queues

danielmi  180  0   n   i   -OEKCVHP  sc  dest: PEOPLE.D.DANIELMI.NOTES
                                     home: PEOPLE.D.DANIELMI
                                     mail: PEOPLE.D.DANIELMI.MAIL

NRCS-A$

```

The first line of the display lists the traits; the other lines list specific values for the traits.

The following is the interpretation of the sample command results previously displayed:

- User danielmi has a read rate (rr) of 180.
- The keyboard preference (kb) is 0.
- The user is a superuser (su). The n means “news superuser.” A minus (-) would appear if the user is not a superuser.
- The edit mode (m) is insert.
- The local only mode (l) is non-local only.
- The traits indicated by SOEKCVHP and sc are explained in the [“User Traits Console Command Summary” on page 268](#). See also [“list u” on page 146](#).
- The users destination queue is PEOPLE.D.DANIELMI.NOTES.
- The users home directory is PEOPLE.D.DANIELMI.
- The users mail queue is PEOPLE.D.DANIELMI.MAIL.

[“User Traits Console Command Summary” on page 268](#) 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.

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 <name> [<option>] <value>[..+flag][..-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:

```
NRCS-A# utraits danielmi readrate 195
```

```
1 user records modified
```

```
NRCS-A#
```

To give him superuser privileges:

```
NRCS-A# utraits danielmi su n
```

```
1 user records modified
```

```
NRCS-A#
```

To take superuser privileges away from him:

```
NRCS-A# utraits danielmi su -
```

```
1 user records modified
```

```
NRCS-A#
```

The blacklist trait is a flag; it is either on or off. You grant flag traits with a plus sign, and you take them away with a minus sign. For instance, to blacklist the user, danielmi, type what appears in bold:

```
NRCS-A# utraits danielmi +b
```

```
1 user records modified
```

```
NRCS-A#
```

To remove him from blacklist status:

```
NRCS-A# utraits danielmi -b
```

```
1 user records modified
```

```
NRCS-A#
```

You can change more than one trait at a time. For instance, to give this user keyboard 3 and make SHOW.RUNDOWN his destination queue, type what appears in bold:

```
NRCS-A# utraits danielmi key 3 dest show.rundown +localonly
1 user record modified
NRCS-A#
```

Changing a User's Password

To change Smith's password to changeme:

1. Enter superuser mode at the console.
2. Type the utraits command using the following format:

```
utraits <username> password <newpassword>
```

For instance,

```
NRCS-A# utraits smith password changeme
1 user records modified
NRCS-A#
```



Since everything you type at the console is recorded in command history, consider using the force command to require the user to change their password the next time they log in. This will help prevent someone from using passwords obtained from the command history.

For users who do not change their passwords, as instructed, use the force console command to require them to change passwords at their next login. The following is the force console command 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, log on as a superuser and type what appears in bold:

```
NRCS-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 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 what appears in bold:

```
NRCS-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 this, use the force console command with the following 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 previous format description.

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` 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, 2009:

```
NRCS-A# force passchg\<05JUL2009 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:

```
NRCS-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 console, type:

```
list password= u
```



Ensure that you include a space between the = and the u.

The following command lists every user who does not have a password.

```
NRCS-A: list password= u
user      rr   kb   su  mode  destination
weisman   0   0    -   o
```

In the previous example, 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 less than character (<) 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 less than character, 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	02JAN2009 10:50am	24JUL2009 9:03am	06JAN2009 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:

```
NRCS-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.



When metacharacters—/ & ; () < > space or tab— or control operators—// && ; ; () / <newline> are used in a command string they must be escaped with a backslash character.

The following are some examples:

`list passchg\<<05JUL2009 u` — This produces a list of all users who last changed their password before July 5, 2009.

`list passchg\>10. u` — This 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, 2009 and July 1, 2009. You can even use this command to check a single user or a group of users.

To the aforementioned, 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, 2009, by typing the following: `list passchg\<<05AUG2009 u`

Information similar to the following appears:

User	DEV	Date Created	Last Login	Last Password
-----	----	-----	-----	-----
mitchell	495	02JAN2009 10:55am	24JUL2009 9:00am	07JAN2009 9:50am

You must specify absolute dates in DDMMMYYYY 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, 2009, and before August 15, 2009, type the following: **list passchg\>01AUG2009\<15AUG2009 u**

Information similar to the following appears:

User	DEV	Date Created	Last Login	Last Password
loyd	433	02JAN2009 11:50am	25AUG2009 9:01am	05AUG2009 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. You can see if they have not changed their passwords in the last 90 days by typing: **list passchg\<90. u mitchell schofield**

Information similar to the following appears:

User	DEV	Date Created	Last Login	Last Password
mitchell	495	02JAN2009 10:55am	24JUL2009 9:00am	07JAN2009 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.

After 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.

User Traits Console Command Summary

The following User Traits Summary table is a summary of iNEWS user traits. The first column shows the trait name as it appears in the Modify User Account dialog box. The second column lists actual commands for assigning or removing each trait from the console. The third column contains an explanation of the trait and an example of utraits console command lines.

Name in Modify User Account dialog box	Utrait Console Command	Definition / Example
Superuser	su n su -	The user has superuser privileges.
Blacklisted	+b -b	The user cannot log in to the iNEWS system.
Simplified	+s -s	The user has limitations, as defined by the Simplified User Settings. Examples: utraits palmer su n utraits palmer su - utraits palmer +b utraits palmer +s
Local Only	localonly	The user can only log in locally. Example: utraits jones +localonly
Overwrite	editmode o	Everything the user enters in a story overwrites the character under the cursor. This is the default. Example: utraits hansen editmode o
Insert	editmode i	Everything the user enters in a story is inserted at the cursor location. Example: utraits hansen editmode i
Home	home	Sets the user's home directory, which usually contains the user's Mail and Notes queues. Example: utraits loyd home people.1.loyd

Name in Modify User Account dialog box	Utrait Console Command	Definition / Example (Continued)
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>utraits loyd dest people.1.loyd.notes</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>utraits loyd mail people.loyd.mail</pre>
Read Rate	<code>readrate</code>	<p>The user's spoken reading rate in words per minute. The system uses the readrate of the designated presenter to determine the audio (air) time of a story.</p> <p>Example:</p> <pre>utraits vandenbergh readrate 180</pre>
User Name	<code>realname</code>	<p>The user's real name (as contrasted with their account's User ID name). For instance, John Chapman may have a User ID of jchapman; his real name is John Chapman.</p>
Media Browse	<code>+vb -vb</code>	<p>Specifies user can search the video server.</p> <p>Example:</p> <pre>utraits palmer +vb</pre>
Connect Services	<code>+c -c</code>	<p>Allows user to connect to any service defined in the system.</p> <p>Example:</p> <pre>utraits smith +c</pre>
Manage Projects	<code>+mp -mp</code>	<p>Enables user to manage projects and facets in the system.</p> <p>Example:</p> <pre>utraits jdoe +mp</pre>

Name in Modify User Account dialog box	Utrait Console Command	Definition / Example (Continued)
Toolbars	+cs -cs	Allows users to create custom toolbars. Example: <code>utraits carver +cs</code>
Color Highlights	+cc -cc	Allows users to configure custom status colors. Example: <code>utraits bagley +cc</code>
Highlight Read Stories	+hr -hr	Specifies that unread stories in the queue are highlighted on the user's screen. Example: <code>utraits davis +hr</code>
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 richards +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 utraits typed in front of it, as shown below. Example: <code>force jordan</code>

Name in Modify User Account dialog box	Utrait Console Command	Definition / Example (Continued)
User Preferences button; Sessions tab; Keyboard field	key ###	Assigns a default keyboard (set of macros) to the user account. Example: utraits vandenberg key 048

Managing Database Traits from the Console

This section provides information on how to view basic or detailed database trait information from the console when needed.

Getting Basic Information

To obtain information about a particular queue or directory from the 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:

```
SRPl0-LlSUGQsXWFiTMm 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 SRPl0-LlSUGQsXWFiTMm; each letter in the header line represents a particular database trait. 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:

```
SRPl0-LlSUGQsXWFiTMm 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-LIsUGQSXWFiTMm sortfield purge dis mbox directory
-R-----I---Q-X----- TITLE      P4.0 D1 -      WIRES.ALL
```

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-LIsUGQSXWFiTMm sortfield purge dis mbox
RUNDOWNS.5PM:
QS-P---I---QA---iT-- TITLE      P0      R1 -
      rg=castread  wg=castwrite  eg=casteditorial  ng=-
queue form=RUNDOWN  story form=STORYFORM
```

The first letter (Q) in the line under the queue name indicates that `RUNDOWNS.5PM` is a queue.

Changing Database Traits from the Console

To change a database trait from the 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 plus (+) turns a mode on. A trait preceded by a minus (-) turns it off. The following is 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 story form and the wires queue panel form to it:

```
dbtraits wires +r sform wires qform wires
```

Changing a Parent Directory Only

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 opposite of the way traits are assigned using the iNEWS Workstation.

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

The following Database Traits Summary table is a summary of iNEWS database traits, (SRPlo-LIsUGQSXWFiTMm) as seen and used in connection with the dbtraits command at the console. These traits might also appear in a GUI format at the iNEWS Workstation.

Name	Location in Display SRPlo-LIsUGQSXWFiTMm	Mode / Option Keyword	Definition / Example
Sequential	SRPlo-LIsUGQSXWFiTMm S-----	+s -s	Lists directories or queues in the order in which they were created. (The default is alphabetical.) To order the RUNDOWN.5PM sequentially, type: <code>dbtraits rundown.5pm +s</code>

Name	Location in Display SRPlo-LIsUGQsXWFiTMm	Mode / Option Keyword	Definition / Example
Read Access	SRPlo-LIsUGQsXWFiTMm -R-----	+r -r	<p>Indicates whether or not stories in the queue are in read-only mode.</p> <p>To set the SHOW.5PM.SCRIPTS queue to read-only mode, type:</p> <pre>dbtraits show.5pm.scripts +r</pre> <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	SRPlo-LIsUGQsXWFiTMm --P-----	+p -p	<p>Indicates whether you can use the print command on all stories in the queue with a single command when system printing.</p> <p>To enable users to print all stories in the SHOW.5PM.SCRIPTS queue with a single command:</p> <pre>dbtraits show.5pm.scripts +p</pre> <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>

Name	Location in Display	Mode / Option	Definition / Example
Queue Being Ordered	SRPlo-LIsUGQSXWFiTMm ---lo-----	-o	<p>This is an indicator, rather than database traits. It indicates the queue's order status.</p> <p>The I 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>The o 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 sorted on the screen with order.</p> <p>The <code>dbtrait -o</code> command 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.</p> <p>To find out who last ordered the SHOW.5PM.SCRIPTS queue, type:</p> <pre>list d-o show.5pm.scripts</pre>
Locked Queue	SRPlo-LIsUGQSXWFiTMm -----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 SHOW.5PM.SCRIPTS queue:</p> <pre>list d-u show.5pm.scripts</pre>
Inverted	SRPlo-LIsUGQSXWFiTMm -----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 WIRES.ALL queue, with the most recent one at the top:</p> <pre>dbtraits wires.all +i</pre>

Name	Location in Display SRPlo-LIsUGQsXWFiTMm	Mode / Option Keyword	Definition / Example
Sorted	SRPlo-LIsUGQsXWFiTMm -----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 SHOW.5PM.SCRIPTS queue:</p> <pre>dbtraits show.5pm.scripts +so</pre>
Update	SRPlo-LIsUGQsXWFiTMm -----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 SHOW.5PM.SCRIPTS should be replaced as new versions are moved or copied to it:</p> <pre>dbtraits show.5pm.scripts +u</pre> <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	SRPlo-LIsUGQsXWFiTMm -----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 Dead queue would normally be unrestricted. To prevent people from opening restricted stories once they are moved to the Dead queue, type:</p> <pre>dbtraits dead +g</pre>

Name	Location in Display SRPlo-LIsUGQsXWFiTMm	Mode / Option Keyword	Definition / Example
Queue Operations Allowed	SRPlo-LIsUGQsXWFiTMm -----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 SHOW.5PM.SCRIPTS queue, type:</p> <pre>dbtraits show.5pm.scripts +q</pre>
Save Version	SRPlo-LIsUGQsXWFiTMm -----N----- -----O----- -----A-----	save - n o a	<p>Determines how many old story versions are retained in each queue. Display values are: N for none, O for original, and A for all.</p> <p>To save only the last version of the People directory (this is the default) use:</p> <pre>dbtraits people save -</pre> <p>To save no version, type:</p> <pre>dbtraits people save n</pre> <p>To save only the original version, type:</p> <pre>dbtraits people save o</pre> <p>To save all versions, type:</p> <pre>dbtraits people save a</pre>
Skip	SRPlo-LIsUGQsXWFiTMm -----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:</p> <pre>dbtraits dead +x</pre>

Name	Location in Display SRPlo-LIsUGQSXWFiTm	Mode / Option Keyword	Definition / Example
Watch Append	SRPlo-LIsUGQSXWFiTm -----W-----	+w -w	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 the iNEWS newsroom computer system, it is crucial that it be assigned to queues receiving data from wire services. For instance, to assign Watch Append to the WIRES.ALL queue: dbtraits wires.all +w
Forms Allowed	SRPlo-LIsUGQSXWFiTm -----F-----	+f -f	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, to assign the forms allowed trait to the SYSTEM.FORMS.R queue: dbtraits system.forms.r +f
Index	SRPlo-LIsUGQSXWFiTm -----i---	+index -index	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, to assign this trait to the WIRES.ALL queue: dbtraits wires.all +index
Text Timing Clocks	SRPlo-LIsUGQSXWFiTm -----T---	+t -t	Turns on the Text Timing Clocks, which appear in the Story Text panel. They are: TTC - Time to cursor BLK - Time of blocked (highlighted) text EST - Estimated read time of entire story To turn on these clocks in the RUNDOWN.5PM, type: dbtraits rundown.5pm +t

Name	Location in Display SRPlo-LIsUGQsXWFiTMm	Mode / Option Keyword	Definition / Example
Monitored	SRPlo-LIsUGQsXWFiTMm -----M-	+m -m	<p>Assigns the monitored trait to a rundown. A queue without the monitored attribute cannot be monitored.</p> <p>The -f flag of mapcheck sets the monitored attribute for all rundowns configured in the SYSTEM.MAP queue:</p> <pre>mapcheck -f system.map.</pre>
Media Index	SRPlo-LIsUGQsXWFiTMm -----m	+mi -mi	<p>Assigns the media indexed trait. This trait is assigned to queues and directories you want to be indexed by the Media Index server, which may be used for search integration with other Avid products.</p>
Sortfield	SRPlo-LIsUGQsXWFiTMm Q----- Title	sortfield	<p>Indicates which form field a sorted queue is sorted by. See “Sortfield” on page 27 for more information.</p> <p>When the sortfield is on, all the stories in the queue are sorted. When used on a large queue, the command prompt and control will not be returned until the entire queue is sorted. This may take a long time dependin on the size of the queue.</p>
Purge Interval		purge	<p>Indicates the “age” stories in a queue can reach before they are purged. See “Purge Interval” on page 29 for more information.</p>
Display Lines/Refresh		dis /+ -refresh	<p>Indicates how many lines of each story in the queue are displayed. See “The dis Column” on page 33 for more information.</p>
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.RUNDOWN queue:</p> <pre>dbtraits show.6pm.rundown qform rundown</pre>

Name	Location in Display	Mode / Option	Definition / Example
	SRPlo-LIsUGQsXWFiTMm	Keyword	
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>dbtraits assignments.today sform future</pre>
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 changeform 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 32 for more information.</p>

Name	Location in Display SRPlO-LIsUGQSXWFiTMm	Mode / Option Keyword	Definition / Example
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 35 for more information.</p> <p>See “Groups” on page 35 for more information.</p>
FTS Index		ftsindex #	<p>Assigns the index base used by the specified directory. All directories default to zero (0). Supported values are from 0 to 49.</p> <p>Here are a few examples:</p> <pre>dbtraits wires ftsindex 1 dbtraits archive.2008 ftsindex 2 dbtraits archive.2009 ftsindex 3</pre> <p>Use the <code>list d-i <directory></code> console command to see which FTS index base is associated with a specified directory.</p>

Sortfield

The format of the sortfield information is:

```
SRPlO-LIsUG-QSXWFiTMm 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:

1. Ensure 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 the `sorted` option (`+so`) at the end of the command to turn on its sorted trait. For instance, type:

```
dbtraits rundowns.5pm sortfield page-number +so
```

Turn on the sorted option (`+so`) before allowing users to re-enter the queue.

Starting the Queue Sort Function from the Console

You can use the 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:

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 appears:

```
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:

1. Type the `dbtraits` command, the queue name you want to sort, and `-o`. For instance:

```
dbtraits rundowns.5pm -o
```

2. Press Enter.

Using either the `dbsort` or the `dbtraits` command turns 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:

```
SRPlo-LIisUGQsXWFiTmM sortfield   purge   dis mbox directory
Q-RP-----I---QsXW----- TITLE      P3.0    D1 -    WIRES.ALL
```

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>]
```

Mailbox Attribute

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:

```
SRplo-LIsUGQsXWFiTmM sortfield    purge dis mbox directory
Q-RP-----I---QsXW----- TITLE      P3.0    D1  -    WIRES.ALL
```

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:

```
SRplo-LIsUGQsXWFiTmM sortfield    purge    dis mbox directory
Q-RP-----I---QsXW----- TITLE      P3.0    D1  -    WIRES.ALL
```

The preview lines trait applies to lines previewed in the Queue panel of the iNEWS Workstation. 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 the dis column. Refresh controls whether the system automatically updates your screen if you view a queue someone else is changing. If the number 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 22 lines of a story at a time, that is the maximum value used with this trait. The minimum value is 0, which would show no preview lines.

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.

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.

Stories in a queue display are usually separated by a blank line. The only exception is any queue that displays one line per story, such as a rundown queue. 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.

FTSindex Attribute

The iNEWS system supports up to 50 separate index bases for fast text searching. This separation allows indexing of more documents while avoiding the index base file size limit of 2GB. The index base is assigned using the dbtraits command and the ftsindex attribute. An ftsindex attribute can be any value from 0-49. All directories default to index 0. The ftsindex attribute is independent of the indexed attribute and determines which FTS index base is used for the data in the queue with that ftsindex value.

For example, to separate index bases, use these commands:

```
dbtraits wires ftsindex 1
dbtraits archive.2008 ftsindex 2
dbtraits archive.2009 ftsindex 3
```



The command list d-i displays the index base used by a directory. When the index base is changed, the directory is automatically removed from its current base and added to the new one.

While FTS handles the separation of the index bases, there is only one ftsindex device on the iNEWS Server, and there is only one ftsidx.exe program on the FTS server.

Interplay Attribute

Interplay servers can be assigned to individual directories using the dbtraits command or the Directory/Queue Properties dialog box. All directories default to Interplay 0.

For example, to assign individual directories, use these commands:

```
dbtraits show.sports interplay 1
dbtraits show.weather interplay 2
```



The command list d-i <directory> displays the Interplay server associated with the directory.

Managing Group Traits at the Console

There are four primary group traits used to apply groups to queues and directories in the database: Read, Write, Editorial, and Notify.

The list d-v console command lists each queue's assigned groups. These groups restrict who can read, write, or delete stories in the queue and indicate who is notified when stories change in the queue. 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
```

Editorial Group

Members of the editorial group can delete, move, and float stories, and add breaks to rundowns.

To set a queue's editorial group, use:

```
dbtraits <queue name> eg <group name>
```

For instance, to assign an editorial group of producers to the queue SHOW.5PM, type:

```
dbtraits show.5pm eg producers
```



If no editorial group is assigned, the relevant permissions apply to the write group to stay consistent with previous functionality.

To list all queues in the database with a particular group assigned as their editorial group, use:

```
list eg=<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 console:

1. Enter superuser mode so that the prompt appears like this:

```
NRCS-A#
```

2. Type:

```
NRCS-A# dbtraits system rg sysop
```

3. To apply changes you made to stories already in the directory, type:

```
NRCS-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 might 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:

```
NRCS-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:

```
NRCS-A# gtraits changegroup suggestions
```

Unless you are a member of the mngmt group, you can not see this queue in the Directory panel or open this queue. You may 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 console, you first must log on as 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 `Phonelists` queue:

At a console, in superuser mode, type:

```
NRCS-A# dbtraits phonelists rg - wg -
```

To apply these read and write group changes to stories already in the `Phonelists` queue, type:

```
NRCS-A # gtraits changegroup phonelists
```

Now, anyone can read and write stories in the `Phonelists` queue.

F The Line Editor, ed

This appendix provides basic information about the line editor, called ed, including procedures for using it to make changes to important system files. These include the system configuration, system profile, and printer or wire profile files.



*Additional information may be obtained by typing **info ed** at the console.*

This appendix contains:

- [Starting ed](#)
- [Editing Commands](#)
- [Saving Changes](#)
- [Quitting ed](#)

Starting ed

When you modify a file, make the same changes to each server's copy of the file, or your system will not run properly. Select *all* servers before starting ed and opening a file for editing to ensure changes you make are applied to each server's copy of the file.

To start ed from the console:

- ▶ 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:

```
NRCS-A$ ed /site/config.test
3624
```

The line editor, ed, returns a number indicating the file size expressed as the number of characters, including spaces and returns. Also, the console prompt, NRCS-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:

```
NRCS-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 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 line editor, ed, responds each time by displaying lines 20, 21, and 22, respectively.

```

19
server      522  action      -   -   ; timed-action
Enter
server      523  action      -   -   ; timed-action
Enter
reslist     701                                ;Data Receiver instance
Enter
wireserver  801  news        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: **18**

To print (to screen) line 10 of the file, type: **10p**

To print (to screen) lines 10-20, type: **10,20p**

To print all lines from the current line to line 20, type: **.,20p**

To print all lines in the file from line 80 to the end, type: **80,\$p**

To display the current line number, type: **.=**

To display the line number of the last line, type: **\$=**

To make the fifth line from the bottom current, type: **\$-5**

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 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:

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: `server 251 action 251 - ;timed action`
- Remember that searches are case sensitive. For instance, searching for `/MONITOR/` does not find `monitor`.
- 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: `/monitor 271/` (monitor and 271 are separated by a space)

The line is: `server 271 monitor 271 - ;Morning Show` (monitor and 271 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
a	<p>The Append command inserts one or more lines after the selected line. For instance:</p> <pre> /websession/ ;websession 900 a inws 511 - gnews - ; inws . </pre> <p>In the above example, the user searches for <code>websession</code>, and ed responds by displaying 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 after the displayed line. 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 data-bbox="479 326 989 458"> 21 terminal 13 9600-8n 1 news - ; c ; not used . </pre> <p>In the above example, the user selects line 21, and ed responds by displaying line 21. The user types c 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 (.) 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> <p>27d - Deletes line 27.</p> <p>30,35d - Deletes lines 30 through 35.</p> <p>40,\$d - Deletes every line from 40 to the end of the file.</p> <p> <i>After each Delete command, as well as after the Append, Change, and Insert commands, the file's lines are renumbered. For instance, if you delete line 27, then delete line 30, you actually delete the lines originally numbered 27 and 31 prior to the first Delete command.</i></p>
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> <p>g/asws/s/asws/inws - Finds all occurrences of asws and uses the Substitute command to replace the first occurrence of asws on each line found with inws.</p> <p>To replace all occurrences on a line, add /g to the end of the command, such as:</p> <p>g/asws/s/asws/inws/g</p> <p>g/websession/d - Finds all occurrences of websession and deletes the lines containing it.</p> <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 before the selected line. For instance:</p> <pre data-bbox="479 326 862 487"> /websession/ ;websession 900 i inws 511 - gnews - ; inws . </pre> <p>In the above example, the user searches for <code>websession</code>, and <code>ed</code> responds by displaying 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 before the displayed line. 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) after the target location. For instance:</p> <p>18m20 - Line 18 becomes line 20. Lines 19 and 20 become 18 and 19.</p> <p>1,5m\$ - Moves lines 1 through 5 to the end of the file.</p>
p	<p>Contrary to the name, the Print command does not send information to a printer. Instead, it prints 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: 10,20p.</p> <p>More examples of the Print command can be found in “Specifying Lines to Edit” on page 292.</p>
s/<old text>/ <new text>	<p>The Substitute command substitutes a specific portion of a line with new text. For instance:</p> <pre data-bbox="479 1277 943 1381"> 3 ;AvidNews Starter config 09JAN10 s/AvidNews/iNEWS ;iNEWS Starter config 09JAN10 </pre> <p>In the above example, the user selects line 3, and <code>ed</code> responds by displaying line 3. The user substitutes the word <code>AvidNews</code> for <code>iNEWS</code> but does not alter anything else on the line. The line editor, <code>ed</code>, confirms the substitution by redisplaying line 3, incorporating the change.</p>

Command	Description & Examples (Continued)
t	<p>The Copy command copies the line specified, and inserts a copy after the target location. For instance:</p> <p>5t10 - Inserts a copy of line 5 below line 10. The copy becomes line 11. The original line 11, if any, becomes line 12, and so forth.</p> <p>20,30t50 - Inserts copies of lines 20 through 30 after line 50.</p>
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 u 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 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 line editor, type **q** and press Enter. For instance:

```
w
3624
q
NRCS-A$
```

In the previous example, the user saved changes first by using the Write command (**w**), 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
```

```
NRCS-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.

Index

Symbols

/etc/hosts 180
 /site/config 181
 /site/dict/dmessages 236
 /site/dict/joblist 234
 /site/dict/keymacros 226
 /site/dict/mcs 230
 /site/dict/messages 211
 /site/dict/queues 221
 /site/dict/smessaging 236
 /site/dict/words 223
 /site/system 193

A

Absolute date, defined 266
 Access restrictions 288
 Action servers
 validation messages 218
 AMS 63
 at command 165
 Avid
 online support 20
 training services 23
 Avid Monitoring Server 62–64

B

Backup tapes 66
 Backups
 policies and procedures 66
 queues 75
 tapeless 69
 to tape 68
 types of 65
 Blocks 49
 defined 50
 bpoll command 166

broadcast command 116
 Broadcast message 116
 bscan command 166

C

Capture command 173
 Category program messages 213
 cccolor 241
 Character mapping 175
 Commands
 Avid personnel only 115
 defining 207
 Configuration file
 example of 181
 configure command 116
 connect command 117
 Connect services 30–38
 dialog examples 33
 dialogs for 31
 Console
 assigning mailboxes to a queue 283
 managing group traits 287
 managing traits 259
 server commands, list of 116
 utraits summary 268
 viewing user traits 259
 Console connect sessions 37
 ctraits command 118

D

D messages dictionary 236
 Data Receiver 106
 Database security 39–47
 Databases
 backing it up 67–77
 backing up queues 75
 backing up tapeless 69

- backing up to tape 68
 - backup policies and procedures 66
 - changing traits at console 272
 - checking for errors 53
 - cleaning 55
 - dbvisit online 58
 - display lines trait 284
 - examining backup tape contents 79
 - free up space 283
 - FTSindex attribute 286
 - getting info. on traits at console 271
 - increase free space 52
 - Interplay attribute 286
 - list date of backups 81
 - list items dumped to tape on specific date 80
 - mailbox attribute 283
 - maintenance procedures 53
 - management 48–58
 - managing traits from console 259
 - monitor free space 49
 - notes on backups 77
 - parent directory 273
 - purge interval trait 283
 - refresh trait for queues 285
 - restoring data 77
 - restoring first-level directory 78
 - restricting access 288
 - search a tape by word and date range 83
 - search a tape by words 82
 - searching a tape 82
 - sortfield trait 281
 - storage units 49
 - story copying processes 50
 - summary of traits 273
 - tracking space usage 51
 - troubleshooting access problems 104
 - troubleshooting locked blocks 106
- date1 and date2 266
- dbclean command 118
- dbclose command 119
- dbdev command 119
- dbdump command 67, 120
- dbfree command 51, 52, 121
 - using 52
- dblincs command 53, 122
- dboriginal command 122
- dbpurge command 123
- dbrestore command 77, 124
- dbserver command 126
 - using 52
- DBServer program messages 213
- dbsize command 119
- dbsort command 126
- Dbtraits
 - messages 220
 - summary of console commands 273
- dbtraits command 127
- dbvisit command 128
- dbvisit online 58
- Dead queue 49
 - unbusy edit-locked stories 104
- delay command 173
- destinationorder 243
- diag command 174
- Dialog boxes
 - Add New User 42
 - Connect to Service 37
 - Manage User Accounts 41
 - Modify User Account, explained 268
- Dialog commands 172
- Dialogs
 - building a 31
 - commands 32
 - defined 31
 - examples 33
- Dictionaries 207
 - case-shifting 228
 - ccucmds 207
 - ccumsgs 207
 - customizing 209
 - d messages 236
 - job list command 234
 - keyboard macros 226
 - mcs 230
 - restoring defaults 211
 - s messages 236
 - utility messages 211
 - words 223
- dictionary command 130
- Directories
 - changing the parent 273
 - hiding SYSTEM directory 288
 - removing restrictions 290
- Dis column 284

disablecommandline 255
 Disconnect program messages 213
 Disconnects 89–100
 causes of 94
 detecting 91
 overview 90
 recovery overview 94
 recovery procedures 95
 recovery worksheet 99
 types of 93
 diskclear command 131
 diskcopy command 131
 distribution command 166
 doc command 131
 dup command 166

E

echo command 174
 ed command 132, 291
 editing commands 294
 quitting the line editor 297
 saving changes 297
 searching a file 293
 specifying lines to edit 292
 starting the line editor 291
 Editing commands 294
 Editorial group 288
 enter command 132
 Environment variables 239
 cccolor 241
 destinationorder 243
 disablecommandline 255
 for servers 256
 mailllookup 244
 msgmailalert 245
 picolor 246
 registry values 240
 RX_HOT_TO_GO 256
 RXDEBUG= 256
 RXDEBUGFILE 256
 RXNOFAST= 256
 RXNOPRESENTER= 256
 RXSENDFORM= 257
 RXSITECHARSET= 257
 RXSITELISTSZ= 257
 showtimingbar 248

synctoserver 252
 VT compatibility 253
 Errors, checking the database 53
 Escape character 174
 escape command 174
 every command 167
 expect command 175
 extension command 167

F

Facets
 list p command 142
 fast command 167
 Flags 146
 force command 133
 force console command 262
 FTS
 checking iNEWS communications 111
 FTSindex attribute 286
 index bases 286
 FTSindex attribute 286
 Ftsseek messages 219

G

Groups
 group checker messages 216
 managing traits from console 287
 security 47
 grpcheck command 133
 Grpcheck messages 216
 gtraits command 133
 gzip 69

H

help command 134
 heol command 175
 Highwater mark 50
 hogs command 51, 134

I

idiff command 135
 ignore command 167
 ignore-del command 167

iNEWS

- backing up system 65–88
- command references 114
- defining commands 207
- defining messages 207
- dictionaries 207
- Input errors 109
- Interplay attribute 286

J

- Job list command dictionary 234
- Job list commands 165

K

- Keyboard checker, program messages 214
- Keyword checker, program messages 213

L

- Last login messages 220
- Line editor, ed 132, 291
 - starting 291
- Linux commands 115
- list B command 139
- list C command 140
- list c command 140
- list command 136
- list d command 140
- list g command 142
- list p command 142
- list q command 145
 - syntax 145
- list s command 145
- list sq command 146
- list u command 146
- list u-v command 259
- local command 168
- Login
 - list users logged in 45
 - monitoring from devices 45
 - tracking user activity 44
- logout command 148
- Lowwater mark 50

M

- Macros
 - dictionaries 226
 - keyboard check program messages 215
- Mail, server messages 218
- Mailboxes, listing all queues (at console) 283
- maillookup 244
- mailto command 168
- makemontab command 148
- makeshift command 148
- maketab command 149
- Map check, messages 218
- Map command 175
- mapin command 176
- mapout command 176
- MCS dictionary 230
 - status reports 234
- message command 176
- Messages
 - broadcast 116
 - defining 207
- Monitor server
 - mcs dictionary 230
- Monitoring free space 49
- Monitoring the free list 50
- monitoring, with AMS 63
- move command 168
- msgclean command 150
- msgmailalert 245

N

- netstat -i command 109
- Network services 31–33
- Notify group 288
- number command 168
- nxserver 153
 - D message strings 236
 - S message strings 236

O

- offline command 153
- on command 168
- online command 154
- Online support 20

open command 169
 order command 169
 otod command 154
 Output errors 109

P

pass command 176
 Passwords 40

- change user passwords at console 262
- checking 267
- checking user status 41
- force user to change 42
- listing change by date 265–267
- listing last change 265

 pause command 177
 picolor 246
 poll command 170
 Print server messages 220
 Procedures

- access registry editor 239
- assign user permissions 104
- back up database to tape 68
- change a queue’s sort field 282
- change default dictionary value 210
- change presenter instructions text color 246
- changing user passwords 262
- check server network connections 102
- check user password status 41
- clean database offline 56
- connect to console from workstation 37
- define key used to advance timing bar 248
- define value for VT compatibility 254
- determine if user is logged in 45
- disable command line option 255
- edit dictionary file 209
- enable destination order 243
- enable message mail alerts 245
- enable synchronized timing 253
- export story to local drive 96
- force user to change password 43
- hide groups from e-mail lists 244
- increase free space in database 52
- install SNMP agent 63
- list date of backups 81
- list items dumped on specific date 80
- monitor logins from devices 46

- obtain space usage information 52
- recover system 96
- restore standard translations 211
- search a tape by word and date range 83
- search a tape for stories 82
- search for user’s last login 44
- set up system service and resource 33
- start or restart queue sorting 282
- starting ed, the line editor 291
- troubleshoot power failure 108
- unbusy edit-locked stories in Dead queue 105
- use dblines to log at specified time 54
- use dblines to log output 54
- view all users logged in 45

Projects

- list p command 142

ps command 107
 publish command 170
 Purge intervals

- defined 49
- format information 283

 put command 170

Q

Q_UNKNOWN 211
 Queues

- /site/dict/queues 221
- changing the sort field 282
- dictionary 221
- display preview lines in rundown 284
- mailbox attribute 283
- refresh trait 285
- removing restrictions 290

 quiet command 170

R

Read group 287
 Read limitations 288
 reconnect command 155
 Recovery worksheet 99
 Registry editor 239
 Relative date, defined 266
 remote command 171
 remove command 156, 171
 rename command 156

reorder command 156
 replace command 171
 restart command 157
 RGB hexadecimal color chart 247
 rr kb su m SOEKCVP sc 146
 RXDEBUG= 256
 RXDEBUGFILE 256
 RX-ERRs 109
 RXNOFAST= 256
 RXNOPRESENTER= 256
 RXSENDFORM= 257
 RXSITECHARSET= 257
 RXSITELISTSZ= 257

S

S messages dictionary 236
 Save error messages 220
 scan command 171
 searchtape command 82, 157
 Security 39–47
 control access via groups 47
 monitoring logins 45, 45
 privacy warning 47
 procedures and guidelines 40
 protecting SYSTEM directory 288
 Seek server messages 219
 send command 158
 send-del command 171
 sendform command 172
 Servers
 environment variables 256
 print messages 220
 Sessions, establishing 102
 showtimingbar 248
 shutdown command 158
 /site/dict/shift 228
 sitedump command 159
 siterestore command 159
 snews program
 D message strings 236
 S message strings 236
 Sortfield database trait 281
 source command 172
 SRPlo-LIsUGQSWFiTMm, explained 273
 startup command 160
 status command 160

stop command 161, 177
 Storage units 49
 Stories
 unbusy edit-locked 104
 understand copy processes 50
 su command 161
 syncserver 252
 System commands 114
 SYSTEM directory
 restricting access 288
 System files 179
 /etc/hosts 180
 /site/config 181
 /site/system 193
 SYSTEM.CLIENT.WINDOWS 194
 SYSTEM.COLORS 195
 SYSTEM.CONFIGURE 196
 SYSTEM.INTERPLAY 197
 SYSTEM.MAP 198
 SYSTEM.RESOURCE 200
 SYSTEM.WIRES 203
 SYSTEM.WIRES.KEYWORDS 204
 AP sample 205
 AP sample 2 206
 System services 33
 System status, normal 90
 SYSTEM.CLIENT.WINDOWS 194
 SYSTEM.COLORS 195
 SYSTEM.CONFIGURE 196
 SYSTEM.GROUP 133
 SYSTEM.INTERPLAY 197
 SYSTEM.KEYBOARDS 221
 SYSTEM.MAP 198
 SYSTEM.RESOURCE 200
 SYSTEM.SEARCHTAPE queue 82
 SYSTEM.SEEK 207
 SYSTEM.WIRES.DISTRIBUTION 203
 SYSTEM.WIRES.KEYWORDS 204
 AP sample 205
 AP sample 2 206

T

Tape operations 66
 Tapes
 examining contents 79
 list date of backups 81

- list items dumped on specific date 80
- search by word and date range 83
- search by words 82
- searching 82
- timer command 177
- Toolbars, disabling command line option 255
- Training services 23
- Troubleshooting 20, 101–113
 - busy stories 104
 - check process status 107
 - iNEWS to FTS communication 111
 - locked blocks 106
 - network failure 108
 - power failure 108
 - recovery worksheet for disconnects 99
 - wires 106
- TX-ERRs 109
- type command 178

U

- unbusy command 161
- Units 49
- User passwords 40
 - checking status 41
 - force change 42
- Users
 - changing passwords 262
 - establishing a session 102
 - list users logged in 45
 - managing traits from console 259
 - modifying traits from console 260
 - summary of traits 268
 - tracking activity 44
 - tracking login activity 44
 - troubleshooting access problems 104
 - troubleshooting log in 102
 - view account creation information 44
 - viewing traits at console 259
- Utility messages 211
- utraits command 162

V

- validate command 172
- Validation
 - server messages 218

- verbose command 272
- version command 163
- VT compatibility 253

W

- W_LOGTYPES 45
- wait command 178
- wholockedit command 164
- Wire distribution
 - sample system file 203
- Wires
 - program messages 218
 - troubleshooting 106
- Workstation, save error messages 220
- Workstation, troubleshooting 102
- Write group 287
- Write limitations 288



Avid
75 Network Drive
Burlington, MA 01803-2756 USA

Technical Support (USA)
Visit the Online Support Center at
www.avid.com/support

Product Information
For company and product information,
visit us on the web at www.avid.com