



Avid® iNEWS®

Setup and Configuration 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 © 2010 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 v3.3 Setup and Configuration Guide • 0130-30597-01 • Created 11/23/10 • This document is distributed by Avid in online (electronic) form only, and is not available for purchase in printed form.

Contents

| | | |
|------------------|--|----|
| | Using This Guide | 23 |
| | Symbols and Conventions | 23 |
| | If You Need Help | 24 |
| | ...With the Syntax of Console Commands | 25 |
| | Avid Training Services | 26 |
| Chapter 1 | Introduction | 27 |
| | Overview of iNEWS | 28 |
| | System Administrator Tasks | 29 |
| Chapter 2 | The iNEWS Console | 33 |
| | Overview of Hardware | 34 |
| | Starting the Console | 35 |
| | Console Display | 36 |
| | Logging in as a System Operator | 37 |
| | Entering Superuser Mode | 37 |
| | Changing System Administration Passwords | 38 |
| | Exiting the Console | 40 |
| | Troubleshooting a Frozen Console | 40 |
| | Types of Console Commands | 41 |
| | Using Server Commands | 42 |
| | Using Console Control Commands | 43 |
| | Assigning Commands to Function Keys | 44 |
| | Displaying Function Key Assignments | 44 |
| | Selecting One or More Servers | 45 |
| | Zooming in on One Server | 46 |
| | Using Console History | 46 |
| | Logged Console History | 49 |
| | Using the Remote Console | 50 |
| | Executing Remote Commands | 51 |
| | Logging out a Remote Console | 51 |

| | | |
|------------------|---|-----------|
| | The Console Configuration File | 52 |
| | Viewing the Console Configuration File. | 55 |
| | Editing the Console Configuration File | 55 |
| Chapter 3 | Getting Started | 57 |
| | Starting the System | 58 |
| | Shutting Down the System | 60 |
| | Starting a System in Single-Server Mode | 62 |
| Chapter 4 | Users | 65 |
| | Viewing User Accounts | 66 |
| | Modifying User Traits | 68 |
| | User Traits | 69 |
| | Changing a User's Password | 74 |
| | Changing User Preferences | 75 |
| | User Preferences | 76 |
| | Setting up Simplified Users | 83 |
| | Simplified User Settings | 84 |
| | Creating New Users | 86 |
| | Creating a New User Area in News Database | 86 |
| | Adding a New User Account | 88 |
| | Enabling a New User to Receive Mail | 91 |
| | Searching for User Information | 91 |
| | Removing User Accounts. | 95 |
| | The User Manager Account | 96 |
| | The Database Manager Account | 97 |
| | Logging Out All Users | 97 |
| Chapter 5 | The Database: Directories, Queues, and Stories | 99 |
| | Overview of the iNEWS Database | 100 |
| | Restrictions to Directory or Queue Creation. | 101 |
| | Creating a New Directory | 102 |
| | Creating a New Queue. | 104 |
| | Outgoing Mail Queue | 105 |
| | Dead Letter Queue | 106 |
| | Search Queues | 107 |

| | | |
|------------------|---|------------|
| | Viewing Search Queue Information from the Console | 111 |
| | Creating a New Story | 111 |
| | Using Script Templates | 112 |
| | Renaming a Directory or Queue | 113 |
| | Deleting a Directory or Queue | 115 |
| | Recovering a Killed Story | 115 |
| | Viewing Database Traits | 116 |
| | Viewing Information about Stories | 118 |
| | Viewing Who Moved, Duplicated, or Killed a Story | 119 |
| | Database Traits Summary | 122 |
| | Changing Database Traits | 136 |
| | Turning Off the Ordered Trait of a Sorted Queue | 138 |
| | Database Purge Intervals and Limits | 139 |
| | Identifying Locked Queues and Stories | 141 |
| | Types of Locks | 144 |
| | Removing Locks from a Workstation | 146 |
| | Unbusy Stories and Queues | 147 |
| Chapter 6 | Groups | 151 |
| | Overview of Groups | 152 |
| | Viewing Group Information from the Console | 152 |
| | Viewing Group Information from a Workstation | 153 |
| | Creating a New Group | 154 |
| | Step 1 - Choosing a Group Name | 154 |
| | Step 2 - Create New Group at Console | 154 |
| | Step 3 - Creating Group's Membership Story and Specifying Members | 155 |
| | Group Checker | 156 |
| | Group Checker Error Messages | 157 |
| | Renaming a Group | 160 |
| | Step 1 - Change Group Name in Database | 160 |
| | Step 2 - Change Group Name in SYSTEM.GROUPS | 160 |
| | Deleting a Group | 161 |
| | Creating or Modifying Multiple Groups | 161 |
| | Adding Users as Members of a Group | 162 |

| | | |
|------------------|---|------------|
| | Adding Groups as Members of Other Groups | 163 |
| | Avoiding Recursion | 163 |
| | Adding Workstations as Members of a Group | 166 |
| | Combined Permissions and Timeouts | 167 |
| | Group Access and Usage Restrictions | 169 |
| | Group Traits for the Database | 170 |
| | Read Group | 171 |
| | Write Group | 172 |
| | Notification Group | 172 |
| | Editorial Group | 173 |
| | Restricted Group | 174 |
| | Restricting Both Reading and Writing | 174 |
| | Transferring Group Assignments | 174 |
| | Hiding Queues and Directories | 175 |
| | Creating a Mail Alias | 176 |
| | Mail Aliases for Other Machines or the Internet | 176 |
| Chapter 7 | Keyboards and Macros | 179 |
| | Types of Macros | 180 |
| | Creating a Keyboard | 181 |
| | Creating Macros | 183 |
| | Using the State Keys in Macros | 187 |
| | Repeating Macros | 188 |
| | Notes of Caution for Creating Macros | 189 |
| | Keyboard Checker | 189 |
| | Testing the Keyboard | 192 |
| | Assigning a Default Keyboard to a User Account | 193 |
| Chapter 8 | Forms | 197 |
| | Form Guidelines | 198 |
| | Creating Forms | 198 |
| | Customizing Forms | 201 |
| | Turning on Label Borders | 206 |
| | SYSTEM.COLORS | 207 |
| | SYSTEM.LISTS | 207 |

| | | |
|-------------------|--|------------|
| | Assigning a Form as a Queue or Story Form | 208 |
| | Form Field Types and Definitions | 208 |
| | Standard iNEWS Forms | 220 |
| Chapter 9 | Character Generator Title Entry | 223 |
| | Overview of CG Title Entry | 224 |
| | Title Entry Setup and Configuration | 226 |
| | CG Template Editor | 229 |
| | Edit Title Entry Template Window | 230 |
| | Creating a New Template. | 233 |
| | Using Font PreSets | 239 |
| | Title Entry Security | 241 |
| | Access to CG Template Editor. | 241 |
| | Access to CG Title Entry | 241 |
| Chapter 10 | System Configuration | 243 |
| | Overview | 244 |
| | Making a Backup File. | 245 |
| | Viewing System Files | 245 |
| | Licensing iNEWS Components | 247 |
| | Configuration File | 249 |
| | Editing the Configuration File | 252 |
| | Testing the Site Configuration File After Alteration | 253 |
| | Incorporating Configuration Changes. | 253 |
| | Hosts File | 254 |
| | System Profile Files | 254 |
| | Changing the System Profile | 255 |
| | Listing Parameter Settings | 256 |
| | System Profile Parameters. | 257 |
| | Devices | 262 |
| | Viewing Information about Devices | 263 |
| | List C Message Columns. | 264 |
| | Adding Devices | 265 |
| | Intersystem Messaging | 267 |
| | Sending Intersystem Messages | 267 |

| | | |
|-------------------|--|------------|
| | Receiving Intersystem Messages | 268 |
| Chapter 11 | Printers | 273 |
| | Local Printing | 274 |
| | Local Printing Dialog Box | 274 |
| | Creating and Using Print Styles | 280 |
| | Local Print Style Options | 280 |
| | Banner Format Options | 286 |
| | Example Style Story | 288 |
| Chapter 12 | Wires | 291 |
| | Adding a Wire – Avid Data Receiver | 292 |
| | Phase 1 | 292 |
| | Phase 2 | 293 |
| | Adding a Wire Profile | 294 |
| | Phase 3 | 295 |
| | Phase 4 | 296 |
| | Wire Profile Options | 296 |
| | Wire Distribution | 297 |
| | The Wire Distribution Story | 297 |
| | Avoiding Hidden Categories | 302 |
| | Mailboxes | 303 |
| | Purge Intervals | 303 |
| | Internationalization | 303 |
| | Setting Up Wire Keyword Searches | 303 |
| | Additional Information about Search Jobs | 306 |
| | Keyword Search Rule Sets | 307 |
| | Keyword Checker Messages | 311 |
| Chapter 13 | Servers | 315 |
| | Overview | 316 |
| | Adding a Server Program to the System | 316 |
| | Job Lists: Queues, Stories, and Commands | 320 |
| | Types of Tasks for Servers | 320 |
| | Adding a Scan Line in a Job List Story | 321 |
| | Defining a Priority Queue | 322 |

| | |
|--|-----|
| Defining an Every Entry Queue | 322 |
| A Server's Command Set | 323 |
| Processing Deleted Stories | 324 |
| Ordered Queues and the Order Command | 325 |
| Mailbox Tasks | 325 |
| Types of Mailboxes | 326 |
| Assigning a Mailbox to a Queue | 328 |
| Timed-Interval Tasks | 329 |
| Example of Timed Interval Tasks | 330 |
| Action Servers | 332 |
| Adding an Action Server | 333 |
| Field Validation | 335 |
| Possible Uses of Validation | 335 |
| Using Field Validation | 336 |
| Validation Job List Commands | 338 |
| Rundown Mirroring | 339 |
| Polling Commands for Action Servers/Tx Links | 340 |
| Configuring Rundown Mirroring | 340 |
| Overlapping Job Lists | 342 |
| Polling Issues Related to Tx Links | 343 |
| Distribution Servers | 344 |
| Distribution Codes | 345 |
| Wildcards and the Destination Queue | 345 |
| Move and Dup Commands | 346 |
| Action Servers or Tx Links | 346 |
| Instructions in the Wire Distribution Story | 347 |
| Matching and Case | 348 |
| Matching and Order | 348 |
| Adding a Distribution Server | 350 |
| Parallel Wire Servers | 353 |
| Adding a Parallel Wire Server | 354 |
| Keyword Servers | 360 |
| Adding a Keyword Server | 361 |

| | |
|--|------------|
| System Servers | 367 |
| Seek Servers | 367 |
| Adding a Seek Server | 367 |
| Fast Text Search Servers | 369 |
| Installing FTS Components on the Windows-based Server | 372 |
| Setting up FTS Components on the iNEWS Servers (Linux) | 374 |
| Batch Indexing | 377 |
| Reindexing (Optional) | 379 |
| Mail Server | 380 |
| Monitor Servers | 381 |
| Checklist: Monitor Server Configuration | 383 |
| Creating a Monitor Server for Each Show | 384 |
| Creating Composite and Event List Queues | 387 |
| Set up Queue and Story Forms | 390 |
| Assigning Forms to Queues | 390 |
| Creating an Entry in the SYSTEM.MAP Story | 391 |
| Updating the iNEWS System Dictionaries | 397 |
| Creating Styles | 400 |
| Using the Monitor Server | 405 |
| Network iNEWS Systems Using RX/TX Links | 407 |
| Sending Story Forms | 407 |
| Setting Automatic Update | 407 |
| Update Trait - Queue Considerations | 409 |
| Changing Queue Order | 410 |
| Adding Rx/Tx Links | 410 |
| Chapter 14 iNEWS Community | 419 |
| Configuring iNEWS for Community | 420 |
| Syntax of the ctraits Command | 422 |
| Viewing Remote Systems or Community Sessions | 423 |
| Restrictions | 423 |
| Local and Remote SYSTEM.MOS-MAP | 424 |
| Removing a System from Community | 424 |
| Connection Issues | 425 |

| | | |
|-------------------|---|-----|
| Chapter 15 | Web Publishing and Access | 429 |
| | Web Publishing | 430 |
| | Setting up Txnet to Send HTML | 430 |
| | The HTML Export Template | 432 |
| | Sample HTML Export Template | 441 |
| | Publishing iNEWS Stories to the Web | 446 |
| | Web Access | 446 |
| | The Web Server | 446 |
| | Logging in via Web Access | 447 |
| | Web Access Story Templates | 447 |
| | Web Access Directory and Queue Templates | 450 |
| | Web Access Configuration | 461 |
| Chapter 16 | iNEWS Projects | 465 |
| | Overview of Projects and Facets | 466 |
| | Setting up the iNEWS Database for Projects | 467 |
| | Creating Projects | 468 |
| | Creating Facets | 471 |
| | Associating Stories with Projects or Facets | 473 |
| Appendix A | Command References | 475 |
| | Programs Invoked by iNEWS | 476 |
| | Commands Used by Avid Personnel Only | 476 |
| | Linux Commands Used in iNEWS | 477 |
| | Console Control Commands | 478 |
| | Console Server Commands | 482 |
| | broadcast | 482 |
| | configure | 482 |
| | connect | 483 |
| | ctrails | 483 |
| | dbclean | 484 |
| | dbclose | 484 |
| | dbdev | 485 |
| | dbdump | 485 |
| | dbfree | 486 |

| | |
|------------------------------------|-----|
| dblines | 487 |
| dboriginal | 488 |
| dbpurge (Superuser conditional) | 488 |
| dbrestore | 489 |
| dbserver | 491 |
| dbsort | 491 |
| dbtraits | 492 |
| dbvisit | 493 |
| dictionary | 495 |
| diskclear (Superuser only) | 495 |
| diskcopy | 496 |
| doc | 496 |
| ed | 496 |
| enter | 497 |
| force (Superuser only) | 497 |
| grpcheck | 497 |
| gtraits (Superuser only) | 498 |
| help | 498 |
| hogs | 499 |
| idiff | 499 |
| list | 500 |
| list B | 502 |
| list C | 503 |
| list c | 504 |
| list d | 504 |
| list g | 505 |
| list p | 506 |
| list q | 508 |
| list s | 508 |
| list sq | 509 |

| | |
|---|-----|
| list u | 509 |
| logout | 511 |
| makemontab. | 511 |
| makeshift (Super user only) | 511 |
| maketab (Superuser only) | 512 |
| msgclean | 513 |
| offline | 517 |
| online | 517 |
| otod | 517 |
| reconnect | 518 |
| remove | 519 |
| rename (Superuser only) | 519 |
| reorder | 520 |
| restart | 520 |
| searchtape | 520 |
| send | 521 |
| shutdown | 521 |
| sitedump (Superuser only) | 521 |
| siterestore (Superuser only) | 522 |
| startup | 523 |
| status | 523 |
| stop. | 523 |
| su | 524 |
| unbusy | 524 |
| utraits (Super user only) | 524 |
| version | 525 |
| wholockedit. | 526 |
| Job List Commands | 527 |
| at | 527 |
| bpoll | 527 |

| | |
|-----------------|-----|
| bscan | 527 |
| distribution | 527 |
| dup | 528 |
| every | 528 |
| extension | 528 |
| fast | 528 |
| ignore | 528 |
| ignore-del | 529 |
| local | 529 |
| mailto | 529 |
| move | 529 |
| number | 529 |
| on | 530 |
| open | 530 |
| order | 530 |
| poll | 531 |
| publish | 531 |
| put | 531 |
| quiet | 531 |
| remote | 531 |
| remove | 532 |
| replace | 532 |
| scan | 532 |
| send-del | 532 |
| sendform | 533 |
| source | 533 |
| validate | 533 |
| Dialog Commands | 533 |
| capture | 534 |
| delay | 534 |
| diag | 534 |
| echo | 535 |
| escape | 535 |

| | |
|--|------------|
| expect | 535 |
| heol | 536 |
| map | 536 |
| mapin | 537 |
| mapout | 537 |
| message | 537 |
| pass | 537 |
| pause | 538 |
| stop | 538 |
| timer | 538 |
| type | 539 |
| wait | 539 |
| Appendix B System Files | 541 |
| /etc/hosts | 542 |
| /site/config | 543 |
| /site/system | 555 |
| console.cfg | 556 |
| SYSTEM.CLIENT.WINDOWS | 558 |
| SYSTEM.CONFIGURE.301-ACTION | 559 |
| SYSTEM.MAP | 560 |
| SYSTEM.RESOURCE | 562 |
| SYSTEM.WIRES.DISTRIBUTION | 565 |
| SYSTEM.WIRES.KEYWORDS | 566 |
| SYSTEM.WIRES.KEYWORDS-AP | 567 |
| SYSTEM.WIRES.KEYWORDS-AP2 | 568 |
| Appendix C Standard Dictionaries | 569 |
| Using Dictionaries to Define Messages and Commands | 570 |
| Customizing Dictionaries | 572 |
| Changing Default Dictionary Values | 572 |
| Restoring Dictionary Defaults | 574 |
| Utility Messages Dictionary (/site/dict/messages) | 575 |
| DBServer Program Messages | 576 |
| Disconnect Program Messages | 576 |

| | |
|---|------------|
| Category and Keyword Check Program Messages | 576 |
| Keyboard Check Program Messages | 577 |
| Keyboard Check Program Messages for Macros | 578 |
| Grpcheck Messages | 579 |
| Wire Program Messages | 581 |
| Mail Server Messages | 581 |
| Validation (Action) Server | 581 |
| Seek Server Messages | 582 |
| Last Login Messages | 583 |
| Print Server Messages | 583 |
| dbtraits Messages | 583 |
| Save Error (Workstation) Messages | 583 |
| Queues Dictionary (/site/dict/queues) | 583 |
| Words Dictionary (/site/dict/words). | 586 |
| Keyboard Macros Dictionary (/site/dict/keymacros) | 589 |
| Case-shifting Dictionary (/site/dict/shift) | 591 |
| MCS Dictionary (/site/dict/mcs) | 593 |
| Device Types Used by Monitor Servers and Drivers. | 593 |
| Special Strings Recognized by the Monitor Server. | 594 |
| Error Messages for the Monitor Server | 594 |
| Status Reported in Device Status Field. | 596 |
| Job List Command Dictionary (/site/dict/joblist) | 597 |
| D Messages Dictionary (/site/dict/dmessages) | 598 |
| S Messages Dictionary (/site/dict/smessages). | 599 |
| Appendix D Environment Variables | 603 |
| Registry Editor | 604 |
| Environment Variables (Registry Values). | 605 |
| CCCColor. | 605 |
| DestinationOrder | 607 |
| MailLookup | 608 |
| MsgMailAlert | 609 |
| PIColor | 610 |
| RGB Hexadecimal Color Chart. | 611 |

| | | |
|-------------------|---|------------|
| | ShowTimingBar | 612 |
| | SyncToServer. | 616 |
| | VT Compatibility | 617 |
| | DisableCommandLine | 619 |
| | Environmental Variables for Servers. | 620 |
| Appendix E | Managing Traits at the Console | 623 |
| | Viewing User Traits from the Console. | 624 |
| | Modifying User Traits from the Console | 625 |
| | Changing a User's Password | 626 |
| | Listing Users Who Do Not Have Passwords | 628 |
| | User Traits Console Command Summary | 632 |
| | Managing Database Traits from the Console | 635 |
| | Getting Basic Information. | 635 |
| | Getting Detailed Information. | 636 |
| | Changing Database Traits from the Console | 637 |
| | Changing a Parent Directory Only | 637 |
| | Database Traits Console Command Summary. | 637 |
| | Sortfield | 645 |
| | Changing a Queue's Sort Field | 645 |
| | Starting the Queue Sort Function from the Console | 646 |
| | Purge Interval | 646 |
| | Mailbox Attribute | 647 |
| | The dis Column | 648 |
| | FTSindex Attribute. | 649 |
| | Interplay Attribute | 650 |
| | Managing Group Traits at the Console | 650 |
| | Read Group | 651 |
| | Write Group | 651 |
| | Editorial Group | 651 |
| | Notify Group. | 652 |
| | Restricting Access Using Read and Write Limitations | 652 |
| | Removing Directory or Queue Restrictions | 653 |
| Appendix F | The Line Editor, ed | 655 |

| | |
|--------------------------------|------------|
| Starting ed | 656 |
| Specifying Lines to Edit | 656 |
| Searching the File | 658 |
| Searching Tips | 659 |
| Editing Commands | 659 |
| Saving Changes | 662 |
| Quitting ed | 662 |
| Index | 665 |

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 Administration 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.

| Symbol or Convention | Meaning or Action |
|---|--|
| > | 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. |

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 Console Commands

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

To view instructions about using a command:

- ▶ 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 m name] -[s] [block# ...]
```

```
`r' for read only
```

```
`s' for "slow" to eliminate cache usage
```

```
`m' for machine name to disconnect
```

```
`i' to just validate isam files
```



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

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 iNEWS console multiplexor 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.

1 Introduction

- 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.

1 Introduction

2 The iNEWS Console

The iNEWS console multiplexor serves as a “command center” that enables system administrators to monitor and maintain the iNEWS newsroom computer system.

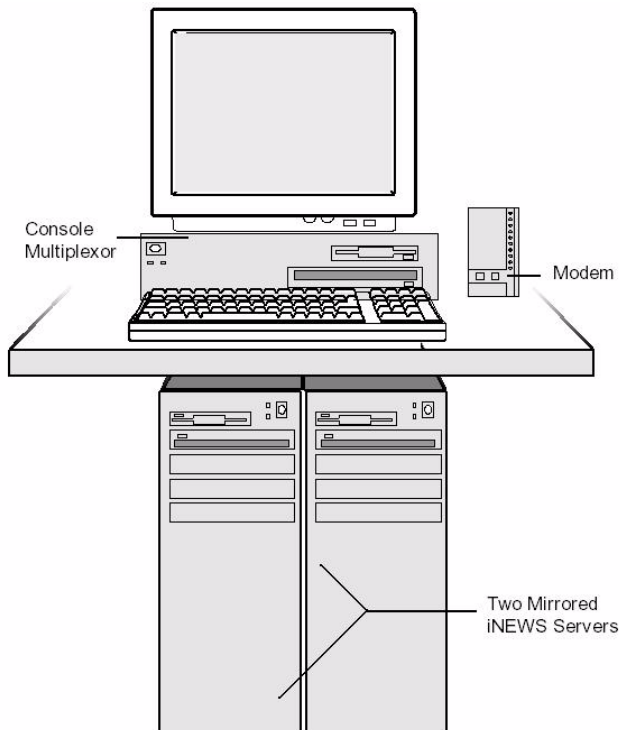
This chapter contains the following main sections:

- Overview of Hardware
- Starting the Console
- Logging in as a System Operator
- Entering Superuser Mode
- Changing System Administration Passwords
- Exiting the Console
- Troubleshooting a Frozen Console
- Types of Console Commands
- Selecting One or More Servers
- Zooming in on One Server
- Using Console History
- Using the Remote Console
- The Console Configuration File

Overview of Hardware

The iNEWS console multiplexor serves as a “command center” that enables system administrators to monitor and maintain the iNEWS newsroom computer system. The iNEWS console multiplexor—commonly known as the console—is a DOS- or Windows-based computer running custom-created Avid software. The console PC connects—via a serial link—to two or more iNEWS Servers and communicates with each server simultaneously.

The following diagram shows a sample layout for a dual-server newsroom computer system.

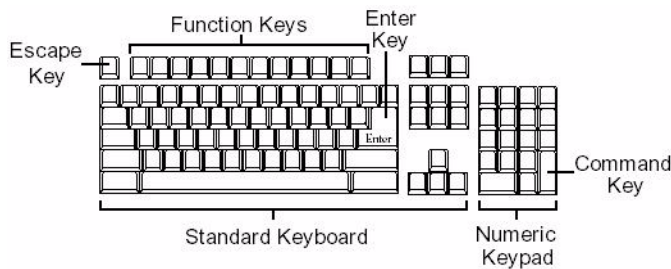


With an external analog modem line, the console also provides a secure remote console feature, which enables a system administrator or Avid Customer Support personnel, if necessary, to do routine maintenance and diagnostic work while away from the computer

room. On newer Windows-based consoles, other remote access software can let administrators access the console securely from inside the network without needing to be at the console. However, Avid Customer Support requires a modem for remote access.

Console Keyboard

While the console multiplexer uses a standard computer keyboard, the Enter key on the numeric keypad is the console's Command key, the Escape key is—by default—used as the Escape character for remote command execution, and the system administrator can preprogram the keyboard's function keys to execute console commands. The following diagram identifies the locations of important keys on the typical console keyboard.



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

Starting the Console

If your console has been turned completely off, it should start the console program automatically when it boots up. If not, it can be started manually.

To start the console program:

1. Do one of the following:
 - ▶ From the operating system prompt, type: **console**.
 - ▶ Double-click `console.exe`.
2. Press Enter.

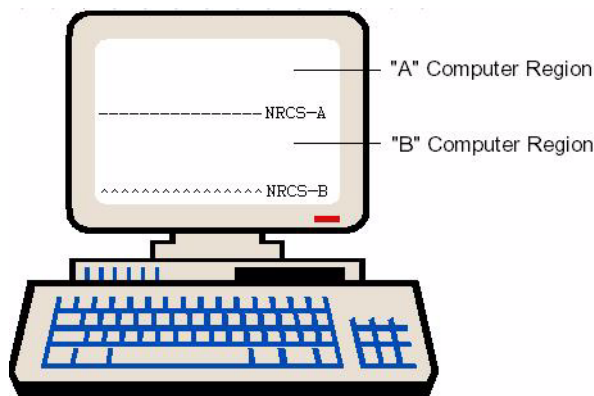
The console display will appear with multiple regions, each region associated with a server and showing the following prompt:

```
login:
```

Console Display


Although the console can control multiple servers, your console has one screen, which is often divided into regions to separate the output from each server. The user can alter the display to show multiple regions or a single region.

The following graphic shows a console display for a dual-server system; the console screen is divided into two regions.



Your console screen has as many regions as there are host servers in your system. In the example, the top region displays the output from server A, and the bottom region displays the output from server B. The wavy lines (^^) to the left of the identifier for server B indicate that it is currently selected or active. Each region will have its own server and command prompts, after which you can enter various commands. See [“Types of Console Commands” on page 41](#) for more information.

To identify which region belongs to which server, the console displays the name of the server that a region represents in that region’s lower right corner.

 *Each server’s name is based on the system ID (typically a station’s call letters) and the server’s name (usually a single letter, such as A, B, or C). Examples in this guide use NRCS as the fictional station and system ID.*

Should data shown in a region extend beyond the size of the screen, the user can temporarily pause the display output. See [“Using Console History” on page 46](#) for more information.

Logging in as a System Operator

System administrators must log in to iNEWS at the console differently than other users who log in at an iNEWS Workstation. For security reasons, system administrators should log out of the system when not using it.

To log in as the system operator:

1. Select the server(s) displaying the login prompt. See [“Selecting One or More Servers” on page 45](#) for more information.
2. Type: `so`.
3. Press Enter.
4. If your system has a password for this account—and most do—then type in the password when prompted. To keep the password confidential, the console does not display what you type.



The system operator password is set during installation of the iNEWS Server software on the iNEWS Servers. For information about changing the system operator password, see [“Changing System Administration Passwords” on page 38](#).

5. Press Enter.

The console’s server prompt will look similar to this:

```
NRCS-A$
```

Entering Superuser Mode

An administrator can take special system privileges as a console superuser when he or she needs to use more powerful—and therefore potentially more dangerous—commands. Once the administrator is done with these more powerful commands, he or she can give up the privileges without logging out. Giving up the privileges (by exiting superuser mode) helps prevent mistakes and provides better security.

The console’s server prompt is the visual indicator for whether you are logged in as a system operator or have enter superuser mode. The server prompt for a system operator login ends with a dollar sign (\$). The server prompt while in superuser mode ends in a pound sign (#).

The system operator prompt looks like this:

```
NRCS-A$
```

2 The iNEWS Console

The superuser prompt looks like this:

```
NRCS-A#
```



If a command example in this guide shows the superuser prompt—ending in a pound sign (#)—you must be in superuser mode to use the command.

A password is required for taking superuser privileges. This password is the same as the root user password, which is set during installation of the Linux operating system on the iNEWS Servers. See [“Changing System Administration Passwords” on page 38](#) for more information.

To take superuser privileges:

1. If you are not already logged in, then log in as a system operator, by typing: `so`.
2. Enter superuser mode by typing: `su`.
3. Press Enter.
4. Type the superuser password at the password prompt. To keep the password confidential, the console does not display what you type.
5. Press Enter.

After you entered the password correctly, the console shows that you have superuser privileges by changing the dollar sign (\$) at the end of the server prompt to a pound sign (#). If you enter an incorrect password, the console displays an error message and returns you to a system operator prompt.



To prevent users from typing unauthorized commands, never leave the console unattended when logged in with superuser privileges. You should enter superuser mode only when you need to type a superuser command, and give up the privileges immediately after typing the command.

To give up superuser privileges and return to the system operator prompt:

- ▶ Press Ctrl+D.

The console shows that you are a system operator by changing the pound sign (#) at the end of the console’s server prompt to a dollar sign (\$).

Changing System Administration Passwords

When logging in to the console as either a system operator or superuser, a password is needed. These system administration passwords are typically set by Avid Customer Support technicians during the installation of either the iNEWS Server software or the Linux operating system (OS). However, they can be changed later by system administrators at the console.



Changing the superuser password also changes the Linux root user password, which is set during the Linux OS installation, must be more than six characters, initially.

Keep a confidential record of password changes. Knowing the passwords is critical. If you forget your passwords, the operating system might need to be reinstalled from scratch by Avid Customer Support technicians.

For more information on changing the password for the remote console, see [“Editing the Console Configuration File” on page 55](#).

To change the system operator password:

1. Log in as a system operator, using the current password.
2. At the server prompt, type the password command, as shown: NRCS-A\$ **passwd**
3. Press Enter.
4. Type the current password, and press Enter.
5. When prompted, type a new password, and press Enter.



If the system operator password is fewer than six characters or is based on a word in the dictionary, the system will issue a “BAD PASSWORD...” message, but it will accept such passwords. The system will not accept a blank password.

6. When prompted to confirm, retype the new password, and press Enter.

To change the console superuser password:

1. Enter superuser mode, using the current password.
2. At the superuser prompt, type the password command, as shown: NRCS-A# **passwd**
3. Press Enter.
4. Type the current password, and press Enter.
5. When prompted, type a new password, and press Enter.
6. When prompted to confirm, retype the new password, and press Enter.



If the password does not match, the system displays an error message. Start over by retyping the new password. Also, if the superuser password is fewer than six characters or is based on a word in the dictionary, the system will issue a “BAD PASSWORD...” message, but it will accept such passwords. The system will not accept a blank password.

7. Press Ctrl+D to leave superuser mode.

The pound sign (#) at the end of the console’s server prompt will change to a dollar sign (\$).

Exiting the Console

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

- You need to edit the console's configuration file. See [“The Console Configuration File” on page 52](#) for more information.
- The console is frozen, and you are unable to unfreeze it using the methods described in [“Troubleshooting a Frozen Console” on page 40](#).

To exit the console:

1. Press the Command key.
2. Press Ctrl+E.
3. When `COMMAND^EXIT` appears, press Enter.

The console application closes.



If running on a DOS system, after the console application closes, the prompt for your operating system will appear.

Troubleshooting a Frozen Console

If the servers on your system are not responding to commands and are not displaying messages, potential causes to look for include:

- Ensure that you or someone else has not stopped scrolling. If you suspect this problem, restart scrolling.
- Ensure that the server ports have not stopped sending and receiving. If you suspect this problem, restart the sending and receiving from server ports.
- Verify whether static electricity froze one or more of the servers' I/O ports. If you suspect this problem, reset the servers' I/O ports.
- Check for an application program that will not stop running. If you suspect this problem, stop the program and log in again to the console.

Here are several possible procedures you can do to identify and resolve the problem.

To check and restart scrolling:

1. If that is the case, XOFF is displayed under the region where scrolling has stopped.
2. Press Ctrl+Q to start scrolling.

To restart sending and receiving from server ports:

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



*In the previous two procedures, the XON command is shown being issued in two ways: first, by pressing Ctrl+Q and second, by pressing the Command key and typing **x**. Either way will accomplish the exact same thing.*

To reset the servers' I/O ports:

1. Select the affected server(s).
2. Press the Command key.
3. Type: **r** (for reset).
4. Press Enter.

To stop a runaway program:

1. Select the affected server(s).
2. Hold down the Control key and press the backslash key (Ctrl+\).

This stops the program on the selected server(s).

If this does not stop the program so that the console displays a system prompt, suspend the program by pressing Ctrl+Z—or whatever the suspend (sus) character is. The default is Ctrl+Z.

You can use the jobs command to list suspended programs and kill suspended programs by using the kill command along with a percent sign (%) and the job number, such as: **kill %1**.

If the console still does not respond, exit the console program and restart it as described in [“Exiting the Console” on page 40](#) and [“Starting the Console” on page 35](#).

Types of Console Commands

You can type two kinds of commands at the console:

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

2 The iNEWS Console

A detailed list of commands is provided as an appendix to this guide. See [“Console Control Commands” on page 478](#) and [“Console Server Commands” on page 482](#) for more information. Also, see [“If You Need Help” on page 24](#) for more information on syntax of console commands.

System administrators can preprogram the function keys on the keyboard to execute commands, too. For more information about assigning commands to function keys, see [“Assigning Commands to Function Keys” on page 44](#).

Using Server Commands

The iNEWS system will not recognize server commands entered in upper case. Type server commands at the console in lower case after the prompt for the server to which you want the command sent. For instance, if you want to send a command to server A, type the command after the server prompt associated with server A.



The console displays each server’s prompt based on the system ID and the server’s name, separated by a hyphen. Examples in this guide use NRCS as the system ID and single letters—such as A, B, or C—as the server name.

For instance, the server prompt might appear like this: NRCS-A\$

The appearance of the server prompt varies, depending on how the system administrator is logged in to the console. See [“Logging in as a System Operator” on page 37](#) and [“Entering Superuser Mode” on page 37](#) for more information.

An example of a server command is the `list sessions` command—or `list s` command—which when sent to an iNEWS Server will return information about who is logged in.

To simultaneously send a server command to multiple servers:

- ▶ Type the server command followed by a pound sign (#).

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

For instance, when you type `connect #` the command `connect a` is sent to server A, `connect b` to server B, and so forth.

To stop a server command:

- ▶ Press Delete.

If that does not work, press Ctrl+C—the stty interrupt character. Doing so will interrupt the command’s execution. As a last resort, pressing the Control and Backslash keys (Ctrl+\—the stty quit character) will stop the command’s execution. Avid recommends you should attempt to “interrupt” before trying to “quit” the execution.

If the server sends a message while you are typing a command, the console stops displaying your keystrokes to display the message. However, it continues to record what you type. After it has displayed the message, then the console will display the data you typed in its entirety.

If you are interrupted by a console display or have mistyped a command, you can cancel the entire command line and start over by pressing Ctrl+U.



The character used to issue the command that cancels an entire command line—known as the “kill” character—may be customized, using the stty command. For instance, to set it to the “at” character, you would type: `stty kill @`.

To reset it to the default Ctrl+U, type: `stty kill ^U`.

The same customization capability applies to the “intr” or “interrupt” character, which is used to interrupt a running program. For instance, type: `stty intr ^?` to set it to the Delete key. To reset it to the default Ctrl+C, type: `stty intr ^C`.

To clarify, the ^U and ^C —shown above— mean to hold the Shift key down and press the 6 key, to get the ^ character, and then the U or C letter key respectively. The same applies to the ^?. You can view these settings by entering the server command: `stty -a`.

Using Console Control Commands

The Enter key on the numeric keypad is the console’s Command key. When you press it, the console displays the command prompt—also called the *command line*—from which you can type in commands.

The command prompt appears as shown:

```
COMMAND^
```

To enter console control commands:

1. Press the Command key.
2. Type the console control command.

Some of the most common console control commands are those used to activate a region of the console display—by selecting an iNEWS Server(s)—and those commands used to view a record of messages displayed on the console, known as the console history.

2 The iNEWS Console

These commands are explained in “[Selecting One or More Servers](#)” on page 45 and “[Using Console History](#)” on page 46. For a complete list of console control commands, see “[Console Control Commands](#)” on page 478.

3. Press Enter.



If you make a mistake when typing a command, use the Backspace key to move the cursor back and then type over the error.

Assigning Commands to Function Keys

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

To assign a command to a function key:

1. Press the Command key.
2. Type the name of the function key you want to use, followed by the equal sign (=) and the command the key is supposed to execute.

The following example assigns the command of choosing server A to the F1 key:

```
f1=c a
```

3. Press Enter.

To assign a command sequence to a function key—that is, include the Command and Enter keys in the definition—use the open brace ({) to represent Command, and the close brace (}) to represent Enter.

For example, to program the complete command sequence (press Command key, type computer command to select all servers, and press Enter) to function key F10, you would type: `f10={c *}`

To change a command assigned to a function key:

- ▶ Assign a new command to the key.

To delete a function key’s command assignment:

- ▶ Assign a null value to the key.

Displaying Function Key Assignments

To avoid inadvertently overwriting a function key’s command assignment, you can check whether a command has been assigned to a specific function key.

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

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

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

Selecting One or More Servers

On the console, you can select one server or multiple servers at the same time. For instance, some commands must be executed on all servers simultaneously, so on a two-server system, you must select both server A and B before typing in the command.

To select only server A:

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

To select both the A and B servers:

1. Press the Command key.
2. Do one of the following:
 - ▶ Type: `computer ab`
 - ▶ Type: `c ab`
3. Press Enter.

To select all servers:

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

Selecting all servers enables you to send a command to all of them simultaneously. When you select all servers, each server region's bottom line changes to a wavy line of caret symbols (^^^). Only one cursor appears, usually in the top region; however, the console commands you type appear simultaneously in each region of the console screen.

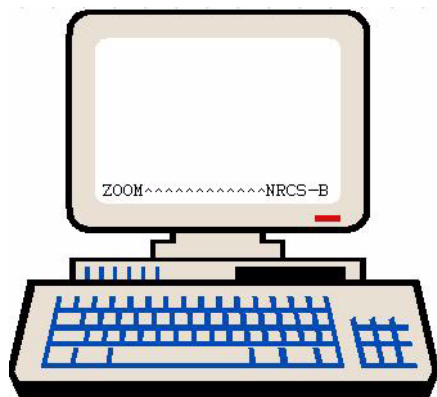
Zooming in on One Server

In addition to the computer command, you can also use the zoom command to select a server. Unlike the computer command, which operates in multiple region mode, zoom selects one server at a time and devotes the entire console screen to that server (or region).

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

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

The following graphic shows the console screen after “zooming in” on server B’s region.



To restore the screen to its former multiple region state, use the `computer` command to select any server. It does not matter which one you select; the console screen will revert to a multiple region display, and the region for the selected server will have the cursor.

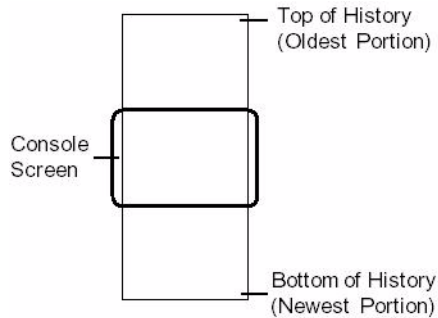
Using Console History

When output from a server command consists of more lines than can fit in a region, lines will be scrolled off the screen. The console maintains a history buffer containing data—known as console history—that has appeared on the screen.



The system can also be configured to log console history for viewing at a later time.

You can go scroll through the console history to review prior activity on any of the iNEWS Servers.



The top of the buffer contains the oldest information; the bottom of the buffer contains the most current. To view this data in screen-sized chunks, you must pause the screen display.

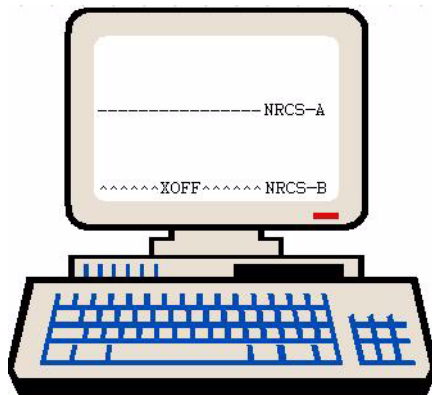
To pause the screen display:

- ▶ Press Ctrl+S. The console screen temporarily stops scrolling.



Pressing Ctrl+S does not have any effect if you are using a remote console. A remote console is connected by a modem dial-in from an external location. See “Using the Remote Console” on page 50 for more information.

When the screen is full of text, XOFF appears at the bottom of the selected region, as shown:



To manually resume scrolling:

- ▶ Press Ctrl+Q.

2 The iNEWS Console

Even if you do not press Ctrl+Q, the console automatically resumes scrolling after a pause of 60 seconds.

To view recent history on a particular server:

1. Zoom in to the server whose history you want to review.
2. Press the Command key.
3. Do one of the following:
 - ▶ Type the `up` command, which moves the console screen up one line in the history.
 - ▶ Type the `down` command, which moves the console screen down one line in the history.
 - ▶ Type the `top` command, which moves the console screen to the top of the history.
 - ▶ Type the `bottom` command, which moves the console screen to the bottom of the history.

If you follow the `up` or `down` commands with a number, the console screen will move up or down that number of lines in the history. For instance, `up 10`, moves up—or back in the console history—10 lines. If you follow the `up` or `down` commands with a word, the console screen will search from the current position backward or forward for that word, and if found, moves to the line containing that word. The commands are not case sensitive.



The pound character (#) is a wild card that can be included in the word to be searched for with the `up` and `down` commands. It is also used to search for numbers. For instance, `up 10#` is a numerical search, and is not the same as `up 10`, which will move up 10 lines in the console history.

4. Press Enter.

The `print` command may also be used to view lines of console history. When the command is followed by a number, the screen will display that number of lines of history, starting from the current position. When the `print` command is followed by the word, `all`, the screen will display all lines from the current position scrolling to the bottom of the history, or newest line of information.



The `print` command does not send console history text to a printer. The command for printing lines of history to a printer is `list`, followed by either a number or the word, `all`. See “Console Control Commands” on page 478 for more information.

Logged Console History

You can configure iNEWS to log console history to disk for later review. The logs are written to the hard drive on your console PC, traditionally in the C:\Console directory or on a Windows-based computer at: C:\Program Files\Avid\Console.

The logs are named as shown in the following table:

| File for server A | File for server B | Contains the following information |
|-------------------|-------------------|------------------------------------|
| log.a1 | log.b1 | Most recent console history |
| log.a2 | log.b2 | Old console history |
| log.a3 | log.b3 | Older console history |
| log.a4 | log.b4 | Oldest console history |

The log.a1 file is the most recent console history; you can also view it as shown in prior examples using the up command.

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

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

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



You can use the view console control command to view the log files without editing or exiting to DOS, but view only lets you start at the top of the file and scroll down. You cannot move back or search for words. See “Console Control Commands” on page 478 for more information.

Both the presence or absence of disk logging and the size of the log files can be configured in the console.cfg file. See “The Console Configuration File” on page 52 for more information.

Using the Remote Console

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

This section explains the requirements for dialing in over a modem line, what you can expect to see on the remote screen, and how to execute commands remotely. It also provides procedures for logging out from a remote console and logging out a remote user from the main console.

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

- Eight data bits
- No Parity
- One stop bit
- 9600 baud rate

To connect as a remote console:

1. Dial in to the console.

To prevent unauthorized people from dialing in to the console, remote access is protected with a password. When you dial in, you see a `PASSWORD` prompt on the screen. If you do not see the prompt immediately, pressing Enter should display it.

2. Enter the password when prompted.

After you type the correct modem password, the console connects you to the first server listed in the console configuration file, usually server A.



The modem password is set in the console's configuration file. See "Editing the Console Configuration File" on page 55 for more information.

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

Executing Remote Commands

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

There are also some differences when using a remote console:

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

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

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



The Escape character, by default, is set to the Escape (Esc) key; however, the character may be customized, using the `escape` command in the following format: `escape <char>`

For instance, to set the Escape character to the keystroke combination of the Control and Y keys (Ctrl+Y), type: `escape Ctrl+Y`.

The Ctrl in the above command means to hold down the Control key while pressing the Y letter key. It is important to remember that the `escape` command can only be used from a remote console and there must be a space after the command.

To reset the Escape character to its default, type the `escape` command, space, then press the Escape key. For instance, type: `escape Esc`.

Logging out a Remote Console

After connecting to the console from a remote location, you should remember to log out.

If you are at the main console and discover that you or someone else has been using the remote console but did not log out when done, you can log out the remote user from the main console.

To log out after you finish using the remote console:

1. Press the Escape character.
2. Type: `l`. The lowercased L in this step stands for logout.
3. Press Enter.
4. When the remote console redisplay the `PASSWORD` prompt, hang up your modem.

2 The iNEWS Console

To log out a remote console user from the main console:

1. Press the Command key.
2. Type: **m**. The lowercased M in this step stands for modem.
3. Press Enter.



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

The Console Configuration File

The console uses information from its own configuration file—a text file called `console.cfg`—to set a number of parameters, such as:

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

This section contains a sample configuration file, defines the console configuration keywords and their parameters, and provides procedures for viewing and editing the file.

Sample Console Configuration File

Here is a sample console configuration file:

```
computer
hostess 3f8
irq 4
label NRCS-A
name A
speed 9600
;
computer
hostess 2f8
irq 3
label NRCS-B
name B
speed 9600
;
computer
hostess 3e8
irq 4
label NRCS-C
name C
speed 9600
;
```

```

modem
hostess 2e8
irq 3
password mux
speed 9600
timeout 0:05

```

The configuration file consists of a list of keywords, such as name and label. Most keywords are followed by parameters, such as a and NEWS. The keyword modem identifies the start of the modem (remote console) section. The keyword computer identifies a server section. Each server and the modem must have its own section in the console configuration file.

Console Configuration Keywords

Here are the configuration keywords and their parameters if they have any.

| Keyword & Parameters | Description |
|--------------------------------|---|
| computer | Indicates the beginning of a server section. Must appear at the top of each server section on the configuration file. |
| hostess <port-address> | Indicates which port address (in hex) the console uses to communicate with a particular device (that is, a server or the modem). This information, which is dictated by the hardware, was placed in the configuration file when your system was installed and should not be changed. |
| irq <interrupt-request-number> | To get the attention of the console, each device (that is, the servers and the modem) connected to the console must have an interrupt request number defined. Typically, these are shared: COM 1 and 3 use the same number, and COM 2 and 4 use the same number. The irq keyword tells the console which interrupt request to expect from each device. This information, which is dictated by the hardware, was placed in the configuration file when your system was installed and should not be changed. |
| label <region-name> | Defines a label that the console uses to identify each server's region of the console screen. The label can be up to 15 alphanumeric characters long, and is typically the system ID. |

| Keyword & Parameters | Description |
|--|---|
| log <drive:filename> [server(s)] [max size] | <p>Enables disk logging. No matter what filename (for example, log) you designate here, the system always uses extensions like A1, A2, and B1, as indicated in “Logged Console History” on page 49. If you do not follow the filename with a list of the servers for which you want history to be recorded, the console records history for all servers. If you omit a log size it defaults to 16,384 bytes.</p> <p>This example creates a log file on the C drive in the same location as the console application (console.exe) for all servers, and its maximum size is 40,000 bytes: <code>log c:log * 40000</code></p> |
| modem | Indicates the beginning of a modem section of the configuration file. |
| name <computer-name> | Names the server described in that section of the configuration file. Each server must have A, B, C, or D as its name. |
| password <password> | The modem password that must be typed when someone logs in at a remote console. The password can have up to eight alphanumeric characters and is case sensitive. |
| portaddress <port> | Selects the DOS address (in hex) the console should use to communicate with a particular device, such as a modem. This information which is dictated by the hardware, is placed in the configuration file during installation and should not be changed. Each device must have a port address defined in its section. The portaddress keyword is used instead of hostess when your system has four servers and a modem. In that case, the modem must use com1 as its port. The port parameter should always be defined as 3f8. |
| speed <baud-rate> | Sets the baud rate for communication between the console and the server. The console ports always communicate at 9600 baud, eight data bits, no parity, and one stop bit. |
| timeout <minutes:seconds> | <p>Allows you to set a time-out value for any modem connection. The system automatically logs out a modem connection if there is no activity for a specified amount of time. For instance, a value of 6:00 would automatically log out a modem connection after six minutes of inactivity. This keyword, which should only be used in the modem section, provides added protection should a user forget to logout from a modem connection to the console.</p> <p>With a value of 0:00—the default value—the feature is disabled, which means the system will not log out a modem connection regardless of inactivity length. The maximum value is 546 minutes and 7 seconds (546:07).</p> |

Viewing the Console Configuration File

You can view the console's configuration file from the console itself or, on a Windows-based PC, open it in a text editing program, such as Wordpad, to view it.

To view the console.cfg at the console:

1. Zoom in on one server, such as server A.
2. Press the Command key.
3. Do one of the following:
 - ▶ Type: `view console.cfg`.
 - ▶ Type: `v console.cfg`.
4. Press Enter to display the first line in the console configuration file.
5. Continue to press Enter to scroll through the file.

Editing the Console Configuration File

Probably the only modification you will ever need to make to the console's configuration file is to change the modem password, used when logging in from a remote console.

To edit the console.cfg file:

1. Open the file using Microsoft's program, Wordpad.
2. Save the file after you complete your modifications.



If the console configuration file is stored on a DOS PC, you need to use a DOS editing tool, such as `edit` to change it, or copy the file to a Windows-based PC to edit it.

2 The iNEWS Console

3 Getting Started

System Administrators are responsible for knowing how to start up and shut down the iNEWS system. This requires logging in at the console multiplexor, which is done differently than other iNEWS users and provides access to features that other logins do not. Information on using the console, including logging in, is covered in Chapter 2. This chapter provides specifics on starting up and shutting down the newsroom computer system.

This chapter contains the following main sections:

- [Starting the System](#)
- [Shutting Down the System](#)
- [Starting a System in Single-Server Mode](#)

Starting the System

The following procedure shows you how to reboot your servers and synchronize them. This is primarily for dual- or triple-server systems. For steps on starting a system in single-server mode, see [“Starting a System in Single-Server Mode” on page 62](#).



Because the following procedure applies to an entire system that has been shut down, you must perform all the steps on all servers, except where otherwise indicated.

To start your iNEWS system:

1. Power up or reboot servers to the login prompt.

The servers will display the following line ten times—one per second:

Press any key to continue.

If no key is pressed, the bootup will continue normally after ten seconds. If a key is pressed, the system displays a message similar to the following:

```
console.exe
GNU GRUB version 0.97 <638K lower / 522176K upper memory>
+-----+
| Red Hat Enterprise Linux Server <2.6.18-128.1.6.e15> SERIAL CONSOLE |
| Red Hat Enterprise Linux Server <2.6.18-128.1.1.e15>                |
| Red Hat Enterprise Linux Server <2.6.18-128.e15>                    |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, 'a' to modify the kernel arguments
before booting, or 'c' for a command-line.
ZOOM^-----^C - console
```

The default is the SERIAL CONSOLE option, for booting to the iNEWS console. Use the up or down arrow keys to select another option; however, the other options should not be used for any reason, unless instructed to do so by Avid Customer Support personnel. Press Enter to continue.



If the system was not shut down as described in “[Shutting Down the System](#)” on page 60, check the console history for messages indicating that all servers shut down at the same time. Do not connect servers unless you are sure their databases are mirrored. If you cannot find messages indicating simultaneous shutdown, or are otherwise unsure whether the disks are mirrored, call Avid Customer Support for assistance before proceeding. If the servers are not mirrored, it will be necessary to bring the system up as a single-server system and go through the re-mirroring process. See “[Starting a System in Single-Server Mode](#)” on page 62 for more information.



If the system was taken through a normal shut down according to instructions, the databases would still be mirrored and you can continue the normal startup procedure.

2. Select *all* servers.
3. Log in as a system operator by typing: `so`
4. When prompted, type the password.
5. Type: `connect #`

The # character acts as a place holder for each server name, allowing you to send a single command to multiple servers simultaneously. So, typing `connect #` will send `connect a` to server A, `connect b` to server B, and so forth.

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

Messages similar to the following appears:

```
connect successful for NRCS-A, starting servers...
```

```
A is OFFLINE. ID is NRCS.
```

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

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

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

6. (Optional) Check for edit and order locks if you are restarting the system after a power failure. During a power failure, the system might not have had time to remove edit and order locks from the database before shutting down.

When you restart the system, remove these locks.



Checking for edit and order locks might take time depending on the size of the database. In an emergency, bypass this step to get the system running. Go back later and remove locks to provide system access. The system can detect invalid locks and will ignore them.

- a. Select *one* server.
- b. Type: `dbclean -x .`

3 Getting Started

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

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

7. Select *all* servers.

8. Type: **startup**

Information similar to the following appears:

```
Checking free space...
```

```
Sep 10 15:42:51 NRCS msg: System is being configured  
database size 10002352,free 9955000=100%,freelist 1991
```

```
Starting NRCS programs...
```

```
booting device 100
```

```
booting server 130
```

The startup command does the following:

- Causes the master computer (usually server A) to read the configuration file
- Brings each server online so users can log in
- Checks database free space (`dbfree`)
- Starts all devices and utility programs

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



Resources used for iNEWS Workstation, Data Receiver, and rxnet sessions do not print any messages until a workstation establishes a connection.

Shutting Down the System

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

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



Because the system requires that you shut down all servers at the same time, most steps in this procedure are performed on all servers simultaneously. Except where instructed to do otherwise, ensure that you have selected all servers before performing each step. See “Selecting One or More Servers” on page 45 for more information.

To shut down your iNEWS system:

1. Broadcast a message to notify users that the system is being shut down. For instance, type: **broadcast -dl Please log out, system being shut down.**
2. Select *all* servers.
3. Log out all users.
4. Type: **status**

This will display the systems current configuration status; verify that the servers are still connected to each other and running in the normal AB configuration for dual-server systems or ABC configuration for triple-server systems. If they are, you may continue with normal shutdown procedures.



If the servers do not all display the same configuration settings—such as “System is AB”—then the databases on the servers are most likely not mirrored and must not be brought back up using normal startup procedures. Contact Avid before proceeding.

5. Type: **shutdown**

A message similar to the following appears:

```
WARNING! This will stop all devices on this computer, and close the
database.
```

```
To prevent loss of work in progress, 'logout all' first.
```

```
Do you really want to do this (y/n)?
```

6. To continue, type: **y** and press Enter.

The screen appears similar to the following:

```
Do you really want to do this (y/n)? y
```

```
/exc/shutdown: Stopping all devices
```

```
/exc/shutdown: Closing database
```

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

7. Log in as a superuser, by typing: **su**

3 Getting Started

8. When prompted, enter the password.

```
NRCS-A$ su
password:
NRCS-A#
```

9. When the superuser prompt appears, shut down the system by using a form of the init command, as shown:

```
NRCS-A# halt
Broadcast message from root (ttySO) (Fri ...):
The system is going down for system halt NOW!
INIT: Stopping atd: [ OK ]
Stopping keytable: [ OK ]
...
Turning off swap:
Turning off quotas:
Unmounting file systems:
Halting system...
flushing ide devices: hda
System halted.
```

10. Turn off each server.

To start up the system again, follow the procedure in [“Starting the System” on page 58](#).

Starting a System in Single-Server Mode

If the servers were not operating in their normal system configuration, such as AB in dual-server systems, or if they were not shutdown using normal procedures according to instructions in this manual, the databases on the servers may not be mirrored.



If the databases are not mirrored, you must not bring the system up using normal start up procedures as instructed in [“Starting the System” on page 58](#) or you will risk database corruption.

If non-mirrored servers, the system administrator must bring the system up in single-server mode and then go through the database re-mirroring process. This section provides the steps for starting a system in single-server mode.



Contact Avid Customer Support for assistance with triple-server systems. It might be possible to start these systems in a dual-server mode, depending on the circumstances of the shutdown.

To start the iNEWS system in single-server mode:

1. Power up or reboot the server on which you want the system to run.
2. Select that server.
3. Log in as a system operator by typing: `so`
4. When prompted, type the password.
5. Do one of the following:
 - ▶ To connect server A as a single system, enter the following command:
`connect a single=a`
 - ▶ To connect server B as a single system, enter the following command:
`connect b single=b`



The syntax `net=a` or `net= b` is also acceptable. If you have a third server, such as server C, use C in place of A or B. Also, notice in the following sample message that the “System is A” not AB or ABC.

Messages similar to the following appear:

```
Sep 10 16:25:52 inews-a last message repeated 15 times
...
Sep 10 16:26:44 inews-a work:[2034]workserver Hot-to-go
connect successful for NRCS-A, starting servers...
A is OFFLINE. ID is NRCS
System is A. Master is A.
Disk status is OK. The database is OPEN.
NRCS-A$
```

6. (Optional) Check for edit and order locks if you are restarting the system after a power failure. During a power failure, the system might not have had time to remove edit and order locks from the database before shutting down.

When you restart the system, remove these locks.



Checking for edit and order locks might take time depending on the size of the database. In an emergency, bypass this step to get the system running. Go back later and remove locks to provide system access. The system can detect invalid locks and will ignore them.

- ▶ Type: `dbclean -x .`

3 Getting Started

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

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

7. Type: **startup**

```
Checking free space...
```

```
Sep 10 15:42:51 NRCS msg: System is being configured  
database size 10002352,free 9955000=100%,freelist 1991
```

```
Starting NRCS programs...
```

```
booting device 100
```

```
booting server 130
```

The entire newsroom computer system will now be run off the single server. The commands entered on a single-server system for shutting down are the same as those used to shut down a dual-server or triple-server system. To return to a dual or triple-server system, it will be necessary to complete the database re-mirroring process.

4 Users

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

This chapter explains how the system administrator can access and change user account information from any iNEWS Workstation. However, user traits can also be viewed and modified at the console. The procedures for using the console are covered in [“Managing Traits at the Console” on page 623](#).

This chapter contains the following main sections:

- [Viewing User Accounts](#)
- [Modifying User Traits](#)
 - [Changing a User's Password](#)
 - [Changing User Preferences](#)
 - [Setting up Simplified Users](#)
- [Creating New Users](#)
 - [Creating a New User Area in News Database](#)
 - [Adding a New User Account](#)
 - [Enabling a New User to Receive Mail](#)
- [Searching for User Information](#)
- [Removing User Accounts](#)
- [The User Manager Account](#)
- [The Database Manager Account](#)
- [Logging Out All Users](#)

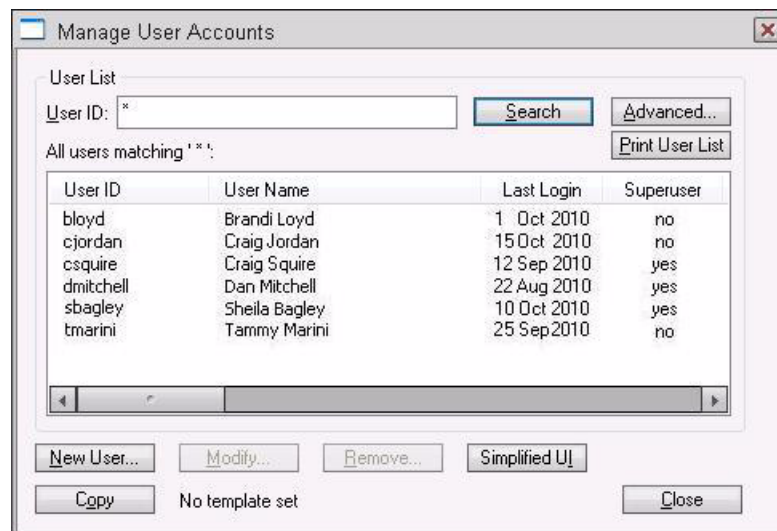
Viewing User Accounts

You must be logged on as a superuser or user manager (umanager) to change user traits. For an explanation of the umanager account and privileges, see “[The User Manager Account](#)” on page 96.

To view traits associated with a particular user account:

1. Select Tools > Options > Users.

The Manage User Accounts dialog box appears.



2. Enter the user name in the User ID field.




An asterisk () in the User ID field will result in all user accounts listed when you click Search or press Enter. To search for all users with names that start with a certain letter, type that letter followed by an asterisk.*

3. Do one of the following:


- ▶ Click Search.
- ▶ Press Enter.

The results of the search appear in the User List field located in the center of the dialog box. The data provided in the field includes: User ID and Name, last time the user logged in, whether the user account has superuser privileges, and so forth. The scroll bar at the bottom of the field may be used to view the rest of the User Account data.

 *The Print User List button will send the User Account data for the results appearing in the User List field to the printer. There is no option to selectively limit what data is printed.*

4. Do one of the following:

- ▶ Double-click the user name in the User List field.
- ▶ Click the name once to select it, and then click the Modify/Display button.

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

The Modify User Account dialog box appears.



The dialog box shows user traits associated with the account you chose, such as the user's name, read rate, and mail queue name. All user traits shown in the various sections of the Modify User Account dialog box are explained in detail in [“User Traits” on page 69](#).

Modifying User Traits

You must be logged on as a superuser or user manager (umanager) to change user traits. The superuser account is an iNEWS user account with superuser type access privileges. It is not the same as the superuser mode used at the console. See [“User Traits” on page 69](#) for more information. For an explanation of the umanager account and privileges, see [“The User Manager Account” on page 96](#).

To modify a user's traits from an iNEWS Workstation:

1. Access the Modify User Account dialog box as explained in [“Viewing User Accounts” on page 66](#).

The Modify User Account dialog box appears.

2. Select or deselect check boxes and fill in the fields in the Queues section of the dialog box as needed. See [“User Traits” on page 69](#) for more information.



You can click the Get from Template button to copy traits from another pre-defined user account. The template must be selected prior to the start of account modification or the button will be inaccessible (grayed out). See [“Adding a New User Account” on page 88](#) for more information.

3. Create or change the password, as explained in [“Changing a User’s Password” on page 74](#).

- Click User Preferences and modify settings, as explained in “[Changing User Preferences](#)” on page 75.
- Click OK to save modifications. The Cancel button closes the dialog box without saving changes.

User Traits

The Modify User Account dialog box divides each user’s traits into sections, such as Type, Edit Mode, Queues, and so forth.

These sections are explained in the following summary of all user traits.

User ID and User Name

The User Name field contains the user’s real name. It should not be confused with the User ID, which the system uses to identify account activity. For instance, a user might have an account with a User ID `dmitche11`; his real User Name is Dan Mitchell, but he will type `dmitche11`, his User ID, to log on to iNEWS.

Type

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



If the check box is selected, the type is applied to that user account.

| Type | Description |
|-------------|--|
| Superuser | A superuser account allows the user complete access to administration features, such as user accounts, the database, the System directory, and connect sessions to the console that controls the servers. |
| Blacklisted | A blacklisted account cannot be used to log in to an iNEWS Workstation. This type is used for special accounts, such as umanager and dbmanager. It is not intended for standard user accounts. Another practical use for this trait is to quickly disable an account of someone leaving or someone who works intermittently. |
| Simplified | A simplified account sets certain access limits, such as the maximum number of iNEWS Workspaces allowed. See “Setting up Simplified Users” on page 83 for more information. |
| Local Only | User accounts flagged as local only cannot be used for session identification by incoming Community sessions. The generic username is used instead. |

Edit Mode

The Edit Mode section’s radio buttons set up the condition of the PC keyboard’s Insert key at log in. The users can still toggle between Insert and Overwrite mode as they work.

| Condition | Description |
|-----------|--|
| Insert | The Insert editing mode, when selected, means if a user types text between two characters, the text is inserted at the cursor position without overwriting the character to the right of the cursor. This is the more typical selection. |

| Condition | Description |
|-----------|---|
| Overwrite | The Overwrite editing mode, when selected, means each character a user types replaces the next character to the right of the cursor as the cursor moves through the text. |

Queues

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

The screenshot shows a dialog box titled "Queues" with three input fields. The "Home" field contains the text "PEOPLE.D.DMITCHELL.NOTES". The "Destination" field also contains "PEOPLE.D.DMITCHELL.NOTES". The "Mail" field contains "PEOPLE.D.DMITCHELL.MAIL".



The actual directory and queues are not created here. The People directory, which can be customized for your environment through a system dictionary, is provided as part of the default database. See “Creating a New User Area in News Database” on page 86 for more information.

| Queue Fields | Description |
|--------------|--|
| Home | The Home field contains the path to the directory (folder) where the Destination and Mail queues are stored in the database file structure. |
| Destination | The Destination field contains the path to the queue provided for the user as a storage location, such as Notes. |
| Mail | The Mail field contains the path to the user's Mail queue, which is where all internal iNEWS mail to that user is kept in the database. For sites that do not use iNEWS mail, set this field to SYSTEM.SHREDDER. Also, even if your site uses iNEWS mail, all template or special accounts (umanager and dbmanager for example) should be set to SYSTEM.SHREDDER. This does not prevent users from sending mail, but it does prevent user accounts from receiving mail. |

Read Rate

The Read Rate is the user's spoken reading rate in words per minute. The average English reading rate is 180 words per minute. The iNEWS system takes the read rate from the user ID named in the story's presenter field to determine the audio (air) time of a story. This also applies to the text timing clocks.

Session Features

There are three sections of the Modify User Account dialog box pertaining to features. The Session Features section defines access to other parts of the system.



If the check box is selected, the feature is applied to that user account.

| Feature | Description |
|------------------|--|
| Media Browse | The Media Browse check box determines access to the Media Browse plug-ins within iNEWS. |
| Connect Services | The Connect Services check box determines access to the Connect to Service dialog box. |
| Manage Projects | The Manage Projects check box determine whether a user can manage projects and facets in the system. |

Configuration Features

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



If the check box is selected, the feature is applied to that user account.

| Feature | Description |
|------------------|---|
| Toolbars | The Toolbars check box determines whether the user can create custom toolbars. |
| Color Highlights | The Color Highlights check box determines whether the user can customize the highlighting status colors in the queue. |
| Highlight Read | The Highlight Read Stories check box specifies that unread stories in the queue are highlighted on the user's screen. The highlight is removed when the cursor is positioned on the story. This feature only applies to stories in a queue with the Watch Appends database trait. |

Queue Features

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

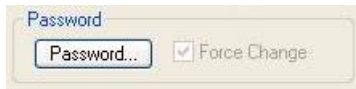


If the check box is selected, the feature is applied to that user account.

| Feature | Description |
|------------------|---|
| Reorder Stories | The Reorder Stories check box determines authority to alter the order of the stories in any queue to which the user has write access. |
| Create/Kill... | The Create/Kill Folders/Queues check box determines authority to create or delete queues and folders (directories) in the database file structure, as seen in the Directory panel of the iNEWS Workspace. |
| Kill All Stories | The Kill All Stories check box determines authority to delete all stories in a single action from any queue to which the user has write access. The data is actually moved from the selected queue to the DEAD queue where it remains (and can be accessed) until purged. |

Password

The Password section has two options with which you can set up or force a user to change an assigned password. See [“Changing a User’s Password” on page 74](#) for more information.



| Option | Description |
|--------------|---|
| Password | The Password button opens a dialog box that you can use to set up or change the password protecting access to the user account. |
| Force Change | The Force Change check box determines whether the user is forced to change the assigned password at the next login. |

User Preferences...

The User Preferences button is used to view and/or modify a user’s preferences, such as keyboard, printer, and confirmation settings. See [“Changing User Preferences” on page 75](#) for more information.

Get from Template...

The Get from Template button is only used when copying the traits of one user’s account to another. See [“Adding a New User Account” on page 88](#) for more information.

Changing a User’s Password

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

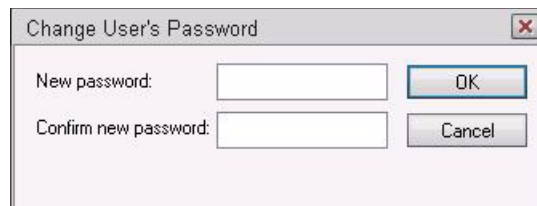


System administrators cannot retrieve a user’s password, only change it.

To change a user's password:

1. Click the Password button in the Modify User Account dialog box.

The Change User's Password dialog box appears.



2. Type the password in the New password field.
3. Confirm the new password by retyping it in the Confirm new password field.
4. Do one of the following:
 - ▶ Click OK to store the password for saving after all modifications to the user's account are done. This will close the dialog box. The process of saving the password is only completed after the OK button on the Modify User Account dialog box is also clicked.
 - ▶ Click Cancel to close the dialog box without saving changes.

Changing User Preferences

System administrators can use the Preferences dialog box to set up default preferences for users. However, the Preferences dialog box is—by default—accessible to users, so they can alter these settings at any time. Users, unless their access is limited by the system administrator, can access the Preferences dialog box to modify their preferences by selecting Tools > Options > Preferences at any iNEWS Workstation. To learn how the system administrator can limit access to this dialog box and its features, see [“Setting up Simplified Users” on page 83](#).

4 Users

To change user preferences:

1. Click the User Preferences button in the Modify User Account dialog box.

The screenshot shows the 'Modify User Account' dialog box. It includes fields for 'User ID' (dmitchell) and 'User Name' (Daniel Mitchell). There are three main sections: 'Type' with checkboxes for Superuser, Blacklisted, Simplified, and Local Only; 'Edit Mode' with radio buttons for Insert and Overwrite; and 'Queues' with text boxes for Home, Destination, and Mail. Below these are 'Read Rate' (195), 'Session Features' (Media Browse, Connect Services, Manage Projects), 'Configuration Features' (Toolbars, Color Highlights, Highlight Read Stories), and 'Queue Features' (Reorder Stories, Create/Kill Folders/Queues, Kill All Stories). At the bottom, there is a 'Password' section with a 'Password...' button and a 'Force Change' checkbox, and buttons for 'User Preferences...', 'Get from Template', 'OK', and 'Cancel'.

The Preferences dialog box will appear, containing several tabs.

2. Modify the preference settings on each tab as needed. The settings are described in detail in [“User Preferences” on page 76](#).
3. Do one of the following:
 - ▶ Click OK to store the preferences for saving after all modifications to the user’s account are done. This will close the dialog box.



The process of saving the preferences is only completed after the OK button on the Modify User Account dialog box is also clicked.

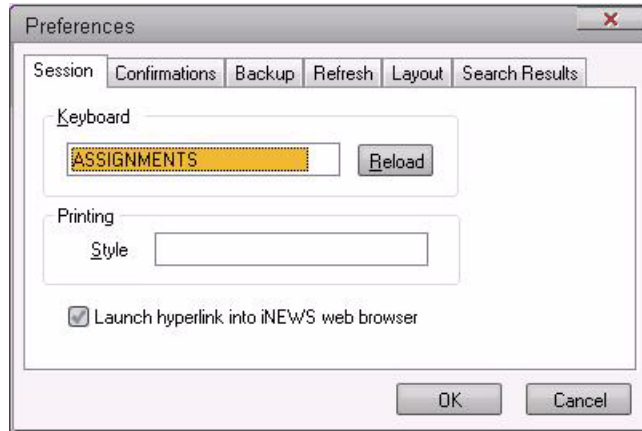
- ▶ Click Cancel to close the dialog box without saving preference changes.

User Preferences

The Preferences dialog box contains several tabs. These various tabs are explained in the following summary of all user preferences.

Session Tab

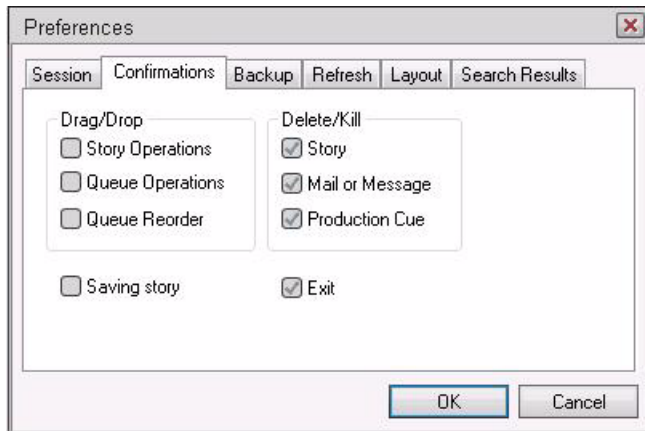
The Session tab has two sections, which system administrators can use to set up default user preferences.





| User Preference | Description |
|-----------------|---|
| Keyboard | The keyboard drop-down list contains a list of keyboards (or sets of macros) that can be assigned to the user account as a default for when the user logs in. The Reload button lets the keyboard assignment take effect without having the user log off and back on. |
| Printing | The Printing drop-down list contains pre-defined Styles that can be assigned to the user account as defaults for when the user prints data from an iNEWS Workstation. |

Confirmations Tab

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

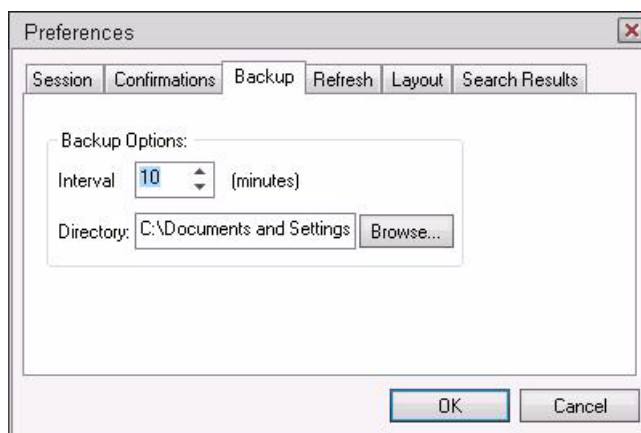


| User Preference | Description |
|------------------|---|
| Saving Story | When Saving Story is checked, iNEWS Workstation will display a confirmation message to save changes before closing an edited story. If you do not select the Saving Story check box, the system automatically saves changes before closing stories. |
| Exit | When Exit is checked, iNEWS Workstation will display a confirmation message when the user attempts to close the iNEWS NRCS program at the workstation. |
| Story Operations | When Story Operations is checked, iNEWS Workstation will display a confirmation message before moving a story when you use the mouse to drag it to its new position. |
| Queue Operations | When Queue Operations is checked, iNEWS Workstation will display a confirmation message before moving all stories in a queue when you use the mouse to drag them to their new position. |
| Queue Reorder | When Queue Reorder is checked, iNEWS Workstation will display a confirmation message before moving a story to a new location in the same queue. |
| Story | When Story is checked, iNEWS Workstation will display a confirmation message before deleting a story or stories. |

| User Preference | Description |
|-----------------|---|
| Mail or Message | When Mail or Message is checked, iNEWS Workstation will display a confirmation message before deleting e-mail or instant messages. |
| Production Cue | When Production Cue is checked, iNEWS Workstation will display a confirmation message before deleting a production cue and its marker from a story. |
| |   <i>A production cue marker (shown at left) appears in the Story Text panel.</i> |

Backup Tab

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



| User Preference | Description |
|-----------------|--|
| Interval | Interval specifies the number of minutes between story backups. The default is 10 minutes. Set the interval to 0 (zero) minutes to turn off the automatic backup feature. |
| Directory | Directory specifies the path name—the location in which iNEWS Workstation should store backup copies of stories. The location should be a directory (folder) on the harddrive of the local PC/workstation. By default, the location is the iNEWS program directory, which is only accessible to computer administrators or power users. Regular Windows users should set this to My Documents. You can type the path in manually, or click the Browse button to select the directory from the Browse dialog box. |

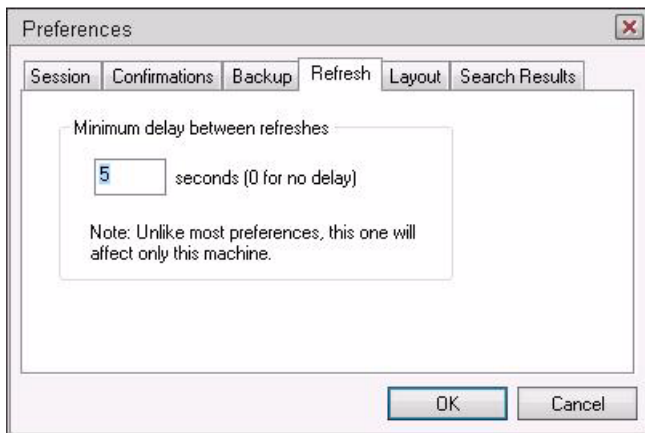
Refresh Tab

The Refresh tab sets the seconds for refreshing the screen at the workstation.



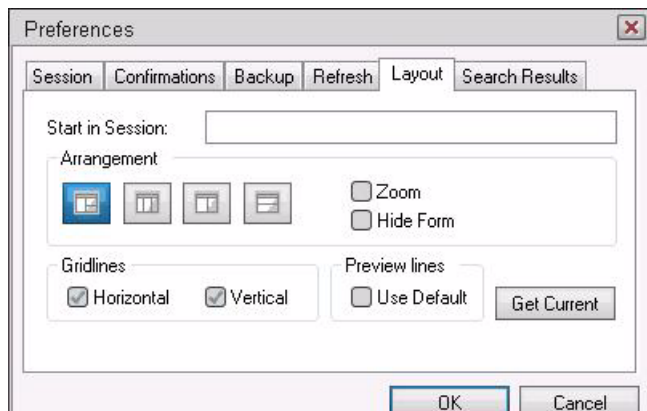
This preference is unique because it only affects the workstation on which it is set.

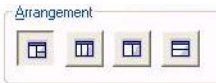
Set the number to zero (0) for instant updating—that is, if you do not want to delay refreshes. A zero delay does require more system and network resources.



Layout Tab

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



| User Preference | Description |
|---|--|
| Start in Session | The Start in Session field specifies the default session that will appear on screen when the user logs in to iNEWS Workstation. |
| Get Current | The Get Current button will reset the preferences on the Layout tab to what was set when the dialog box opened. |
|  | Depending on which Arrangement button is selected, the iNEWS Workstation will display the panels of the iNEWS Workspace accordingly. |
| Zoom | In the Arrangement section, when Zoom is checked, iNEWS Workstation will display the iNEWS Workspace in zoom mode—that is, zoomed into one of the three panels: Directory, Queue or Story. |
| Hide Form | In the Arrangement section, when Hide Form is checked, iNEWS Workstation will display the Story panel with its Story Form panel hidden. The user can choose to show the Story Form panel by selecting the option to show the form from the Story drop-down menu. |
| Horizontal | In the Gridlines section, when Horizontal is checked, iNEWS Workstation will display horizontal gridlines between rows in the Queue panel. |
| Vertical | In the Gridlines section, when Vertical is checked, iNEWS Workstation will display vertical gridlines between columns in the Queue panel. |
| Use Default | In the Preview Lines section, when Use Default is checked, iNEWS Workstation will display the default number of preview lines for each story in the Queue panel as defined by the queue’s properties. When Use Default is not checked, every queue will display only one line of information per story in the Queue panel. |



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

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

4 Users

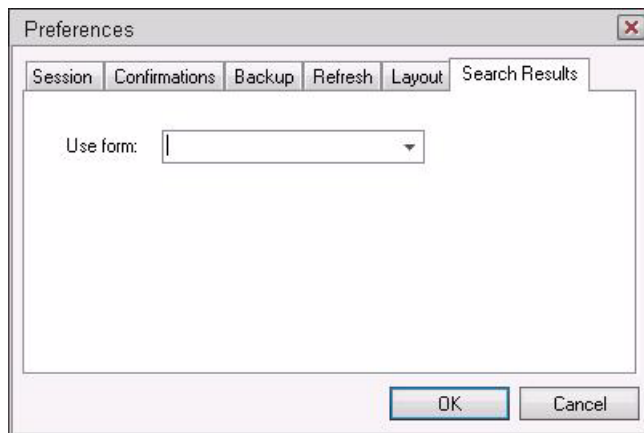
| Slug | Lines | From | Move |
|---|-------|------------|------|
| AM-IA--NewsBriefs 1stAdd 08-30 AM-IA--NewsBriefs 1stAdd 08-30 0380 AM-IA--News Briefs, 1st Add,370 Eds: Moved for PMs siodubdeskaccar12 SIOUX CITY, Iowa (AP) -- Motorists, including some police officers in their squad cars, were stranded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

The second figure shows the Queue panel display without preview lines. This is the view when Use Default is unchecked, unless otherwise specified.

| Slug | Lines | From | Move |
|--------------------------------|-------|------------|------|
| PM-InternetHoax 08-30 0482 | 54 | APp----ar | 8/30 |
| AM-RUN--Michael'sMagic Bjt 08- | 89 | APt----sr | 8/30 |
| AP-WI--ARX-FireWeatherFo 08-30 | 97 | APv2wi-ooa | 8/30 |
| AP-15thNewsAgenda 08-30 1044 | 126 | APv1ntg-dr | 8/30 |
| AP-HurricaneDennisInterm 08-30 | 71 | APv2nwhaor | 8/30 |
| AP-PMPrep-ScoresandSkeds 08-30 | 37 | APv1al-cr | 8/30 |
| AP-HurricaneDennisInterm 08-30 | 69 | APv2nwhaor | 8/30 |
| AP-PMPrep-TV/Birthdays 08-30 0 | 51 | APv1al-cr | 8/30 |
| AX-ABCN1:BabyGiraffe 08-30 017 | 35 | APvabc-xtr | 8/30 |
| AX-ABCN1:DrugSurvey 08-30 0441 | 56 | APvabc-ntr | 8/30 |
| AX-ABCN1:MoreNCDennis 08-30 01 | 42 | APvabc-ktr | 8/30 |
| AM-IA--Bullet-proofVests 08-30 | 41 | APkDia--nr | 8/30 |
| AX-ABCN1:StateDept 08-30 0401 | 55 | APvabc-ktr | 8/30 |
| AM-IA--DryHydrants Bjt 08-30 0 | 57 | APkDia--nr | 8/30 |
| AM-Lebanon-Israel 08-30 0267 | 32 | APp----ir | 8/30 |
| AM-IA--NewsBriefs 1stAdd 08-30 | 47 | APkDia--nr | 8/30 |
| AM-BRF--LightsRecall 08-30 020 | 26 | APp----wr | 8/30 |
| AM-PneumoniaTest 08-30 0493 | 56 | APp----wr | 8/30 |
| AM-IA--MarcianoCrash Bjt 08-30 | 51 | APkDia--nr | 8/30 |
| AP-15thNewsMinute 08-30 0309 | 39 | APv1al-hr | 8/30 |
| AP-PMPrep-Segue 08-30 1112 | 145 | APv1al-cr | 8/30 |
| AX-ABCN1:SETopStory 08-30 0269 | 46 | APvabc-vtu | 8/30 |

Search ResultsTab

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



The iNEWS Workstation will use the form selected from the Use form drop-down list when displaying the results from searches in iNEWS. The forms you can choose from are those created and stored in SYSTEM.FORMS. See [“Creating Forms” on page 198](#) for more information.

Setting up Simplified Users

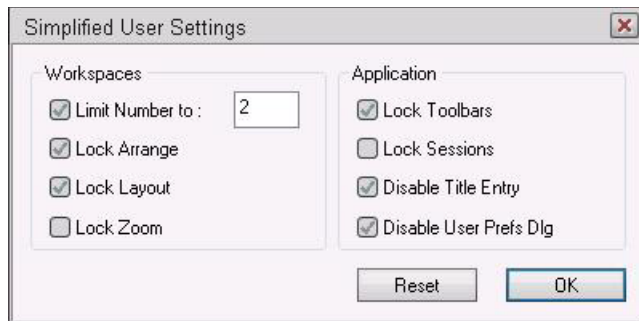
A simplified user is one that has certain limitations pertaining to the iNEWS Workstation. As the system administrator, you can define the limitations and then assign them to users. Only one set of limitations can be defined, which is then applied to all user accounts with the simplified user trait.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

Some of the Simplified User Settings lock the user’s preferences to those defined by the system administrator using the Preferences dialog box. See [“User Preferences” on page 76](#) for more information.

To set up or modify the simplified user limitations:

1. Access the Modify User Account dialog box as explained in [“Viewing User Accounts” on page 66](#).
2. Click the Simplified UI button.

The Simplified User Settings dialog box appears.



The dialog box divides the settings into two sections, which are explained in the next sections of this chapter.

3. Select or deselect check boxes, as required.
4. Click the OK button to save the settings and close the Simplified User Settings dialog box.



Use the Reset button to discard changes and reset the check box settings to what they were when the dialog box opened.

Simplified User Settings

The Simplified User Settings dialog box splits the settings into two sections. These sections are explained in the following summary of all simplified user settings.

Workspaces Section

The Workspaces section of the dialog box provides settings that pertain to the arrangement and quantity of workspaces within the iNEWS Workstation's main window.



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

Application Section

The Application section of the dialog box provides settings that pertain to accessing certain iNEWS features at any workstation.



| Setting | Description |
|---------------|--|
| Lock Toolbars | When Lock Toolbars is checked, iNEWS Workstation will prevent the user from altering the display of toolbars. |
| Lock Sessions | When Lock Sessions is checked, iNEWS Workstation will prevent the user from creating or altering sessions. The user will be locked to sessions created prior to the Lock Sessions being applied to the user account. |

| Setting | Description |
|-----------------------|---|
| Disable Title Entry | When Disable Title Entry is checked, iNEWS Workstation will prevent user access to the Title Entry dialog box, used to enter production cues in stories, and the Edit Title Entry Template dialog box, used to create CG templates for the Title Entry feature. |
| Disable User Prefs... | When Disable User Prefs Dlg is checked, iNEWS Workstation will prevent user access to the Preferences dialog box. The user will be unable to alter user preferences, such as default printer settings. The user will be locked to settings already in place at the time Disable User Prefs Dlg is applied to the user account. See “Changing User Preferences” on page 75 and “User Preferences” on page 76 for more information. |

Creating New Users

To set up new users in iNEWS, you must complete three separate procedures:

- Create areas in the iNEWS database file structure where the user can store notes and receive mail. See [“Creating a New User Area in News Database” on page 86](#) for more information.
- Add a new user account so that your system recognizes the user. This includes setting up the user traits associated with the account. See [“Adding a New User Account” on page 88](#) for more information.
- Enable the new user to receive mail by adding him or her to the appropriate group. See [“Enabling a New User to Receive Mail” on page 91](#) for more information.



To ensure a user has appropriate database privileges, the user should also be assigned to the correct group or groups. For more information, see [“Adding Users as Members of a Group” on page 162](#).

Creating a New User Area in News Database

While a user account will work without this area, it is traditional for each user to have his or her own area in the database to keep notes and to receive e-mail. Usually, these areas are separate queues called Notes and Mail. These queues are kept in a sub-folder—with the user’s account name—in the People directory.

The common practice is to separate the first level of People sub-folders by using the first initial of the user’s last name—otherwise, since the system is limited to 250 folders in the People directory, your site might eventually reach the limit.

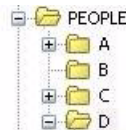
For instance, the following procedures, the Home directory for our sample user, DANIELMI, would be: PEOPLE.D.DANIELMI. The Notes and Mail queues would be: PEOPLE.D.DANIELMI.NOTES and PEOPLE.D.DANIELMI.MAIL, respectively.



You must be logged on to iNEWS NRCS with a user account that has authority to create new directories and/or queues to complete these procedures.

To create a new directory:

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

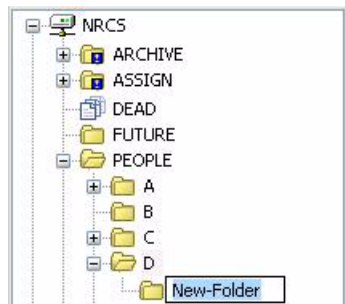


For instance, for user DANIELMI, you would select PEOPLE, then the folder with the alphabetic name corresponding to the first letter of the user's name, such as D. This ensures the new folder will be created in the D directory (folder).

2. Do one of the following:

- ▶ Select Tools > New Folder.
- ▶ Right-click on the folder in the Directory panel, and select New Folder from the context menu.

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



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

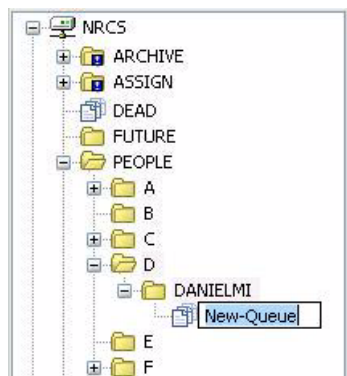
You can now create new queues for the user, such as Notes and Mail queues.

4 Users

To create a new queue:

1. Navigate to and select the folder created to hold the queue you want to create, such as PEOPLE.D.DANIELMI.
2. Do one of the following:
 - ▶ Select Tools > New Queue.
 - ▶ Right-click on the folder in the Directory panel and select New Queue from the context menu.

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



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

Adding a New User Account

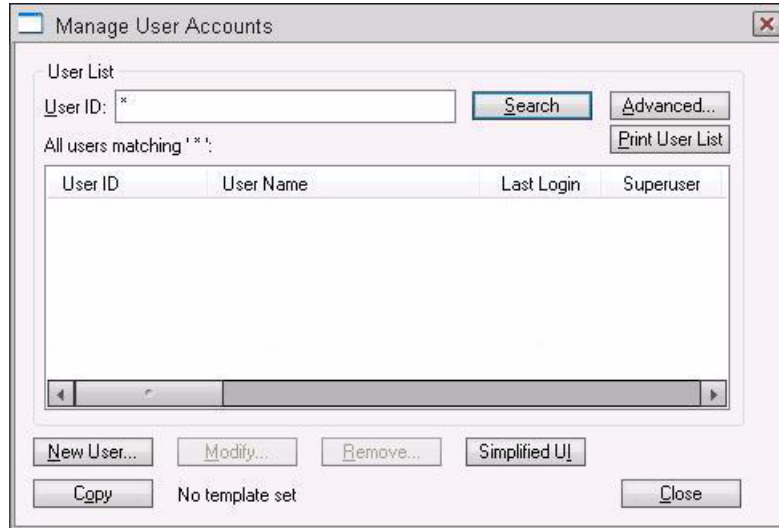
When adding a new user account, you have the option of creating the account from scratch or copying the traits of another user account already in the system. This section covers both options.

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

To define an account as a template for copying to other accounts:

1. Select Tools > Options > Users.

The Manage User Accounts dialog box appears.



If you do not have superuser privileges, which permits access to the Manage User Accounts dialog box, the system will prompt you for the umanager password. If the umanager account does not exist in the system, then access is only allowed to system administrators—that is, those with superuser privileges. See “The User Manager Account” on page 96 for more information. Also, see “User Traits” on page 69 for more information on the superuser trait and its privileges.

2. Search for the user account you want to use as a template for copying user traits. See “Searching for User Information” on page 91 for more information.
3. Select the User ID when it appears in the search results list.
4. Click Copy. The User ID should appear to the right of the button. When no template is selected for copying, the words, “No template set,” appears to the right of the Copy button.

Once a Template User is established, selecting the New User... button will start with the same configuration as the template user, and selecting Get from Template when modifying an existing user will set the configuration to that of the template user.

4 Users

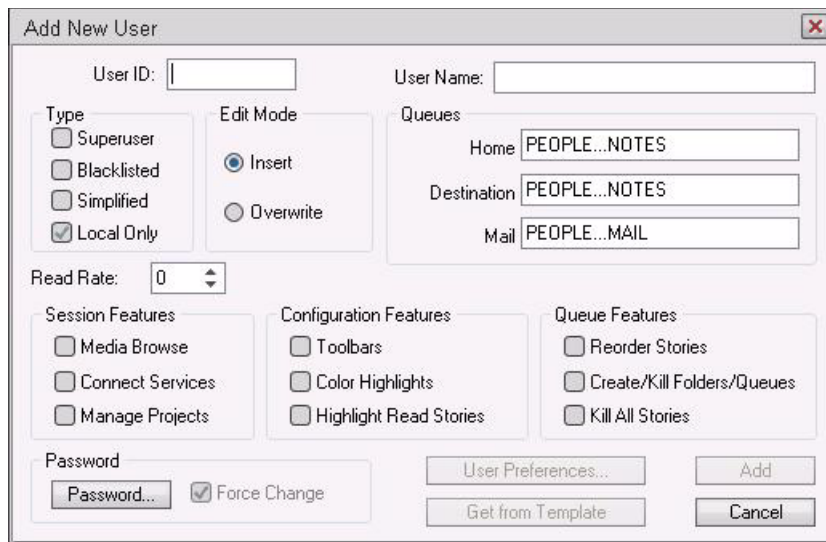
To add a new user account (either from scratch or by copying the user traits):

1. Select Tools > Options > Users.

The Manage User Accounts dialog box appears.

2. Click New User.

The Add New User dialog box appears.



3. In the User ID field, enter the login name of the user account.
4. (Optional) In the User Name field, enter the user's real name.
5. (Optional) Modify the various traits you want to apply to the new user account.



A user's traits can be modified at the time of creation or afterwards from the iNEWS Workstation or from the console. For more details about the various user traits, see "User Traits" on page 69 or "User Traits Console Command Summary" on page 632.

6. Click Add to add the new user account.

Enabling a New User to Receive Mail

This section gives you the basic steps you need to follow to enable a new user to receive mail. If you need more information, refer to the information on groups in [“Adding Users as Members of a Group”](#) on page 162.

To enable a new user to receive mail:

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

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

Searching for User Information

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

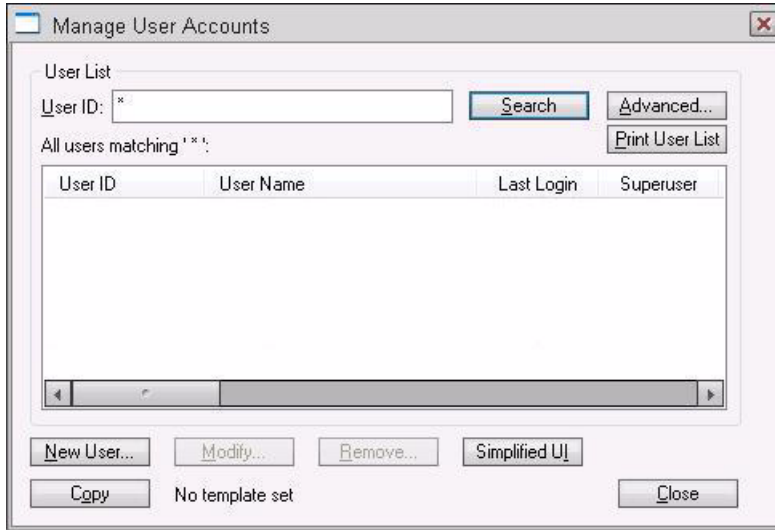
For instance, if you are searching for all user accounts beginning with Dave, type Dave* (no space).

4 Users

To search for information about users:

1. Select Tools > Options > Users.

The Manage User Accounts dialog box appears.



2. Enter the name of the user in the User ID field.
3. Click Search or press Enter.

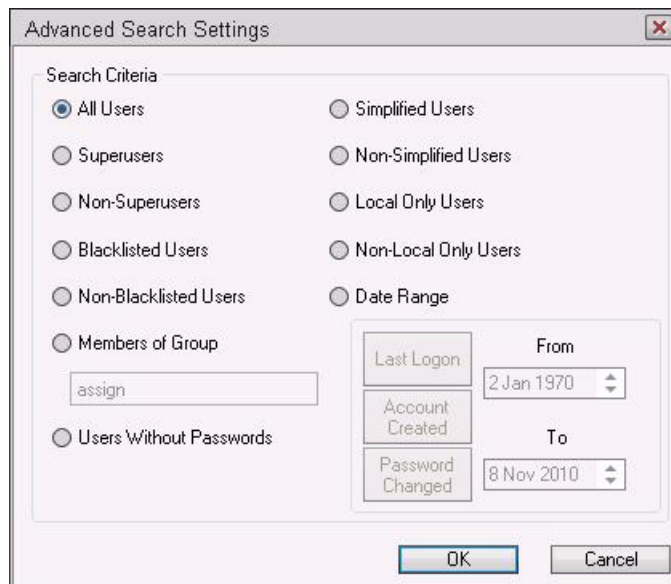
If you search with a wildcard character and the system finds multiple matches, a results box appears listing all “hits”. You can specify one by double-clicking on it; this opens the specified user’s information in another dialog box.

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

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

4. Click Advanced to refine your search for a user.

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



5. Select from the settings to specify additional search criteria. The criteria options are explained in detail below.

| Settings | Description |
|-------------------------|---|
| All Users | Search through all user accounts on the server. |
| Superusers | Confine the search to user accounts with the superuser attribute. |
| Non-Superusers | Confine the search to user accounts without the superuser attribute. |
| Blacklisted Users | Confine the search to user accounts with the blacklisted attribute. |
| Non-Blacklisted Users | Confine the search to user accounts without the blacklisted attribute. |
| Members of Group | Confine the search to user accounts belonging to the security group you select from the drop-down list. |
| Users Without Passwords | Confine the search to user accounts that do not have passwords. |
| Simplified Users | Confine the search to user accounts that have the simplified user trait. |
| Non-Simplified Users | Confine the search to user accounts that do not have the simplified user trait. |
| Local Only Users | Confine the search to local system user accounts only. |

| Settings | Description |
|----------------------|--|
| Non-Local Only Users | Confine the search to non-local system user accounts only. |
| Date Range | Confine the search to dates you specify in the From and To fields and the kind of date range: When the user last logged in When the user account was created When the password changed Specify the date by either clicking the arrow buttons or typing the dates in ddmmmyyyy format. Indicate the day with two digits, the month with three letters, and the year with four digits. |

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

Above the field, iNEWS NRCS will display a brief statement indicating what matched the search criteria, such as:

All users matching '*':

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

Removing User Accounts

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

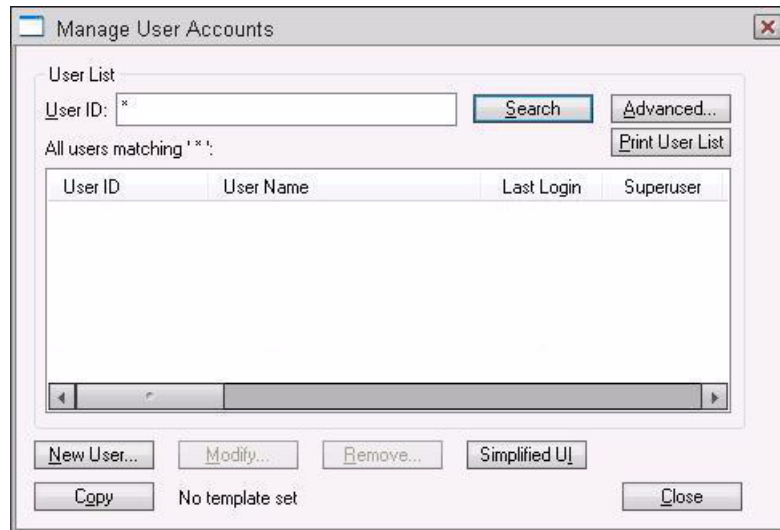


Before removing user accounts from the system, remove the users' names from the stories in SYSTEM.GROUPS. This reduces the potential for mail bounce back errors.

To remove user accounts:

1. Select Tools > Options > Users.

The Manage User Accounts dialog box appears.



If you are logged on as a system administrator, the Manage User Accounts dialog box will appear automatically, following step 1. Otherwise, iNEWS will prompt you for the umanager password (if that account exists in the system) before allowing access to the dialog box.

2. Enter the name of the user in the User ID field.
3. Click Search or press Enter.

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

4. Select the name of the user you want to remove by clicking the name in the User List field.

5. Click Remove.
6. Click OK to remove the user or Cancel to stop the removal.



After removing the user, you will also need to remove the user's Home directory and the Notes and Mail queues by deleting them from the system's database file structure. You can also use the `msgclean` command at the console to remove any of the user's unread messages.

The User Manager Account

A user manager has some special system privileges, but not as many as a system administrator/superuser. For instance, user managers can add, remove, or change any user account, except those with the superuser attribute.

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

To create a user manager account:

1. Create a user account as explained in [“Adding a New User Account” on page 88](#).
2. Give the account a User ID: `umanager`.
3. Assign a password to the account.
4. Make the user manager account blacklisted so that no one can use it to log in to the system.
5. Assign the account superuser status to prevent a user manager (or anyone who does not have superuser status) from changing the account's password.
6. Tell the user manager(s) the ID and password for the user manager account.



For further security, a write-access group should be assigned to `SYSTEM.GROUPS` and only those with user manager privileges should be included in the group. If no write-access group is assigned to `SYSTEM.GROUPS`, then all users who know the `umanager` password can access the Manage User Accounts dialog box by selecting **Tools > Options > Users. After a write-access group is set up, any user managers with nonsuperuser accounts must be included in the write-access group for `SYSTEM.GROUPS` or they will not be allowed access to the Manage User Accounts dialog box. See [“Groups Tab” on page 128](#), [“Adding Users as Members of a Group” on page 162](#), and [“Group Access and Usage Restrictions” on page 169](#) for more information.**

The Database Manager Account

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

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

To create a database manager account:

1. Create a user account as explained in [“Adding a New User Account” on page 88](#).
2. Give the account a User ID: **dbmanager**.
3. Assign a password to the account.
4. Make the database manager account blacklisted so that no one can use it to log in to the system.
5. Assign the account superuser status to prevent a user manager (or anyone who does not have superuser status) from changing the account’s password.
6. Tell the database manager(s) the ID and password for the database manager account.

Logging Out All Users

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

To log out all users:

1. Select all servers.
2. To prevent users from logging in, take the system offline by typing: **offline**
Use **offline silent** if you want to suppress output of messages from new user login attempts.

4 Users

- Use the broadcast command to send a message warning all users that are logged in that they must log out and why. If the system will be shut down, include the time it will be shut down. Here are a few examples:

```
NRCS-A$ broadcast -l WARNING\! System shut down at 12PM
```

```
NRCS-A$ broadcast -l LOG OUT\! System shut down at 5 minutes.
```

```
NRCS-A$ broadcast -dl LOG OUT\! System shut down in 10 seconds.
```



The backslash (\) before the exclamation point (!) is required because the exclamation point is a reserved character in Linux.

The -l (the letter L, not the number 1) after broadcast is to limit the message to local workstations only and not to users connected through Community. You may also add -d after broadcast and before the message if you want your message to appear at the workstation as a popup message. The two can be combined as shown (-dl) to broadcast local popup messages.

- At the specified shutdown time, check the system for any users still logged in by selecting one server and typing: `list s`

A message similar to the following appears:

```
G505 miller      A
G500 allen      B
G507 stevens    A
R801 stevens    A
```

A connect session will show up as an 'R' device. The system administrator must notify them of the shutdown by some other means, such as by telephone.



You must ensure that all users are logged out if shutting down the system. If a user is in a connect session when the system is shut down, the user's workstation stops, the session is disconnected, and any unsaved work is lost. Ensure any connect session users have logged out before you continue the shutdown procedure.

- To stop a connect session, select the appropriate server and then use the following format of the stop command: `stop <device number>`.

For instance, to stop the connect session (R801 stevens A) that was shown in the previous example, type: `stop 801`

- Select all servers.
- Log out all users by typing: `logout all`



If a user is editing a story, the system saves the file and then logs out the user. The logout command will not log out users who are in connect sessions.

5 The Database: Directories, Queues, and Stories

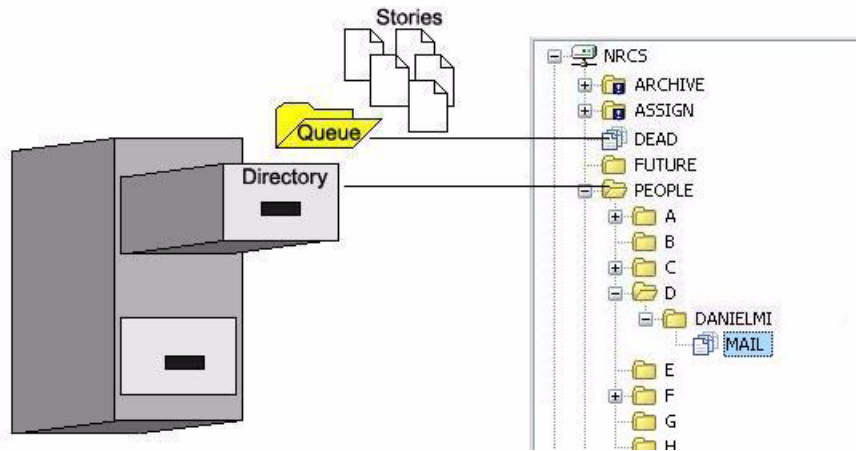
All relevant iNEWS information—except the system software and the Linux files—is stored in the iNEWS database. This database contains stories, queues, and directories or more specifically: scripts, rundowns, e-mail, messages, users, groups and their memberships, and any other kind of information entered into the system. Some database maintenance, such as altering the database file structure and traits, can be done from the console or from any iNEWS Workstation. This chapter focuses on maintenance tasks at the workstation when possible. However, when a task can be done at both the workstation and console, the console information is provided as an appendix in this guide. See procedures for using the console in [“Managing Traits at the Console”](#) on page 623.

This chapter contains the following main sections:

- [Overview of the iNEWS Database](#)
- [Restrictions to Directory or Queue Creation](#)
- [Creating a New Directory](#)
- [Creating a New Queue](#)
- [Creating a New Story](#)
- [Renaming a Directory or Queue](#)
- [Deleting a Directory or Queue](#)
- [Recovering a Killed Story](#)
- [Viewing Database Traits](#)
- [Changing Database Traits](#)
- [Database Purge Intervals and Limits](#)
- [Identifying Locked Queues and Stories](#)
- [Removing Locks from a Workstation](#)
- [Unbusy Stories and Queues](#)

Overview of the iNEWS Database

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



In iNEWS, the database file structure—directories and queues—can be seen (depending on your access privileges) in the Directory panel of the iNEWS Workspace. Directories, also known as folders, are indicated by manila folders, such as the People directory in the previous graphic. Queues are indicated by three overlapping pieces of paper, such as the Dead queue.



If your site uses the Projects feature of iNEWS, tabs will appear in the Directory panel letting you select whether you want to view the iNEWS database file structure via the Directory tab or a list of defined projects via the Project tab. For more information on the Projects feature, see “iNEWS Projects” on page 465.

The scripts, notes, e-mail, news stories, and other kinds of information are all called stories; each story is contained in a particular queue, and each queue in a single directory. A directory can also be contained within another directory in which case it would be called a subdirectory or sub-folder of that directory.

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

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

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

Database traits are used in conjunction with the user traits discussed in “Users” on page 65. For instance, stories in a queue that has a read group specified can be read only by users who are members of that user group.

Restrictions to Directory or Queue Creation

Before you can modify the database traits of queues or directories, the queues and directories must exist in the database. If they do not, you can create them from the iNEWS Workstation. However, there are certain restrictions you should be aware of when creating new directories and queues.

- The total path name of a directory, including the separator characters (.), cannot exceed 60 characters.
- The total path name of a queue, including the separator characters (.), cannot exceed 62 characters.
- Each branch of a path name—that is the name between periods—cannot exceed 20 characters.
- The number of directory levels available is limited to 31.
- You cannot use a space or period in directory or queue names.
- The system has a limit of 250 queues per directory.



The 250 limit also applies to first-level sub-folders in a directory. If you need more than 250 in a directory, such as employee folders, create alphabetic sub-folders on the first level, then place the personal folders in the matching sub-folder. For instance, in a given directory, you could have 26 sub-folders, each with one of the 26 letters of the English alphabet as a name. This enables you to have 250 personal folders in each of the 26 alphabetic sub-folders; that’s enough personal folders for up to 6,500 employees. See “Creating a New User Area in News Database” on page 86 for more information.



While punctuation marks can be used in path names, Avid strongly recommends their use be limited to dashes and underscores only to avoid confusion for FTP clients, like teleprompters, accessing the database. It’s best to limit path names to 0-9, A-Z, hyphen (-), and underscore (_) characters only because using anything else will have consequences for console commands and could affect system functioning.

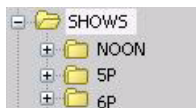
Ordinarily, directories and queues are listed in alphabetical order within their parent directory. However, you can add items to a directory in a different order. For instance, if you had directories for each month in the Futures directory, you would want them to appear in order by month (January, February, and so on). To do this, turn on the sequential database trait for the parent directory before you create the new items. See [“Database Traits Summary” on page 122](#) for more information.

Creating a New Directory

If you are creating a new first-level directory, be sure to select the server rather than a directory or folder as explained in this procedure.

To create a new directory (folder):

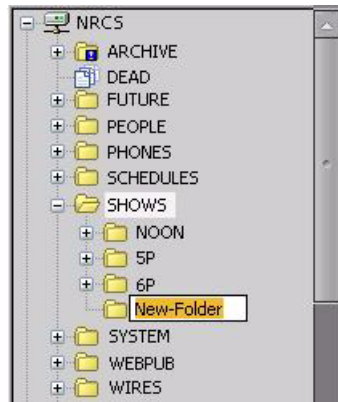
1. Log in as a system administrator unless you have write-access to the parent queue or directory. This ensures that you have full access to the database.
2. Using the database file structure in the Directory panel of the iNEWS Workspace, select the directory under which you want the new folder to be created, as shown in the following example.



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

3. Do one of the following:
 - ▶ Select Tools > New Folder.
 - ▶ Right-click on the directory—or server, if you are creating a new first-level folder—in the Directory panel, and choose New Folder from the context menu.

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



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



Once created, directories cannot be renamed from the iNEWS Workstation. If you incorrectly name a newly created directory, delete it and recreate it with the appropriate name. See “Renaming a Directory or Queue” on page 113 or “Deleting a Directory or Queue” on page 115 for more information.

After you create a directory, you can configure its properties by right-clicking on the queue and selecting Properties. A quick way to apply properties is to use a template. For more information on copying properties from templates, see “Directory/Queue Properties Dialog Box” on page 122.

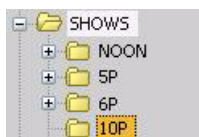
Creating a New Queue

Similar to creating a new directory, you can create a new queue from an iNEWS Workstation.

To create a new queue:

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

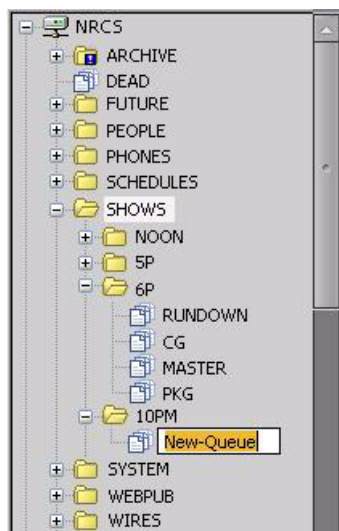
For instance, select the Shows folder, then the 10P folder, if you want to create new queues, such as the Rundown and Master queues for the 10PM show.



The new queue will be created in the folder you select.

2. Do one of the following:
 - ▶ Select Tools > New Queue.
 - ▶ Right-click on the folder in the Directory panel and choose New Queue from the context menu.

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



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



Once created, queues cannot be renamed from the iNEWS Workstation. If you incorrectly name a newly created queue, delete it and recreate it with the appropriate name. See “Renaming a Directory or Queue” on page 113 or “Deleting a Directory or Queue” on page 115 for more information.

After you create a queue, you can configure its properties by right-clicking on the queue and selecting Properties. A quick way to apply properties is to use a template. For more information on copying properties from templates, see “Directory/Queue Properties Dialog Box” on page 122.

Outgoing Mail Queue

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



The outgoing mail queue’s name is defined in /site/dict/queues. When your system was installed, the outgoing mail queue was defined in this dictionary as SYSTEM.MAIL.OUT. To change this name, modify the dictionary entry in /site/dict/queues. The token used to define the outgoing mail queue is Q_MAILOUT. See “System Configuration” on page 243 for more information on editing system files.

Mail always uses the form that is assigned to the outgoing mail queue. See “Mail Form” on page 230 for more information.

After mail arrives in that queue, a utility program known as the mail server processes and sends it to its intended destination. If the outgoing mail queue, SYSTEM.MAIL.OUT, does not exist in the System directory, the mail server cannot distribute e-mail. To create this queue, follow the instructions provided in “Creating a New Queue” on page 104, making these adjustments:

- The queue’s name should be “Out.”
- The queue should be located in the Mail folder, which is in the System directory. If the Mail folder (directory) does not exist, create it. See “Creating a New Directory” on page 102 for more information.

Moreover, the Out queue’s read group must be set properly for the mail server to process mail correctly. See “Read Group” on page 171 for more information. Although not required, you can restrict read permission for the queue SYSTEM.MAIL.OUT to a group that has no

users. Doing so does not interfere with anyone's ability to send e-mail, but it prevents anyone (except superusers) from reading mail in SYSTEM.MAIL.OUT waiting to be processed. See [“Restricting Both Reading and Writing” on page 174](#) for more information.

Also, for your system to notify the mail server when new mail arrives in SYSTEM.MAIL.OUT, that queue must have the same mailbox number assigned to it as the mail server. If the queue does not have a mailbox or has an incorrect one, your system has no way to notify the mail server when there is mail to process. See [“Maintain Tab” on page 130](#) for more information.



The mailbox number is not solely related to e-mail. This database trait lets you link a queue to a server (utility program) so that the system notifies the server program when stories are added to or edited in the queue. See [“Maintain Tab” on page 130](#) and [“Servers” on page 315](#) for more information.

Dead Letter Queue

Your system must also have a dead letter queue, usually called SYSTEM.MAIL.ERROR. This queue is a final destination for any e-mail that your system is unable to deliver to the addressee or return to the sender.



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

If SYSTEM.MAIL.ERROR does not exist, any mail that the mail server cannot deliver or return to the sender is put in the Dead queue. To create SYSTEM.MAIL.ERROR, follow the instructions provided in [“Creating a New Queue” on page 104](#), making these adjustments:

- The queue's name should be “Error.”
- The queue should be in the Mail folder, located in the System directory. If the Mail folder (directory) does not exist, create it. See [“Creating a New Directory” on page 102](#) for more information.

Returned mail may contain sensitive information. Therefore, restrict read permission for SYSTEM.MAIL.ERROR to a group that has no members. Then, only superusers can read mail in the queue. See [“Read Group” on page 171](#) for more information. Examine the queue occasionally to see whether any mail exists.

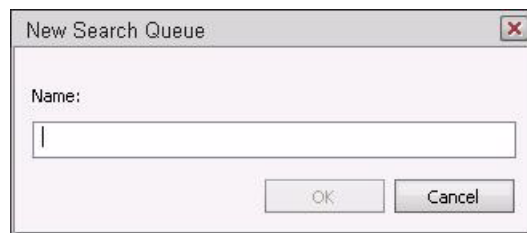
Search Queues

A search queue is a special queue that stores a pre-defined query of indexed queues, along with its own queue form and read group. When users with proper read access open a search queue, a fast text search (FTS) is activated, and the results are delivered to the workspace in the Queue panel instead of the Search Results pane. The results can then be copied, moved, deleted, and edited from the workspace. How fast the results are returned depends on the speed of your FTS system, the size of your index, and the number of areas that is being searched, but the usual timeframe is between one and 15 seconds.

To create a search queue:

1. At an iNEWS Workstation, in the Directory panel, right-click on a folder in which you want the search queue created.
2. Select New Search Queue.

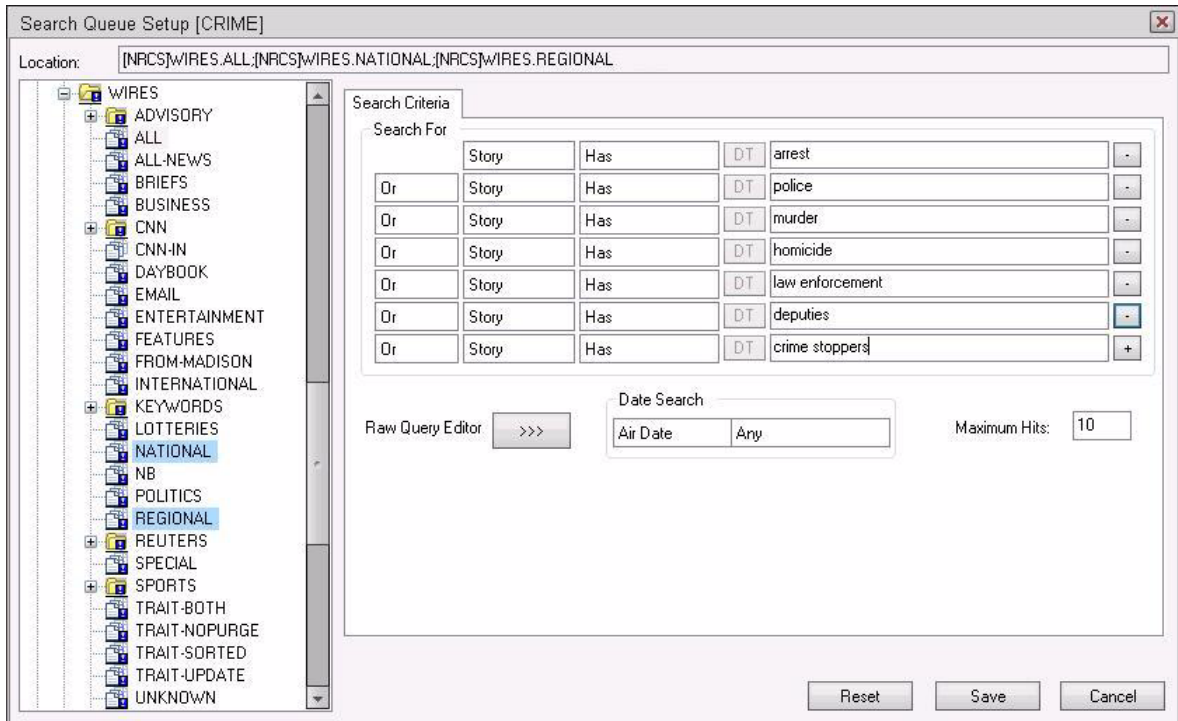
The New Search Queue dialog box appears.



3. Type the name of the new search queue and click OK. The naming restrictions for search queues are the same for those of directories and other queues.

The Search Queue Setup dialog box opens with the name of the search queue appearing in square brackets in the title bar. For instance, CRIME is the search queue name in the following sample image.

5 The Database: Directories, Queues, and Stories



4. Select the indexed queue or queues you want to search from your site or a Community site, if available. To choose multiple queues, press and hold the Control key while clicking on your selections.
5. Use the Search For area of the Search Criteria tab to set the filters for your search. Otherwise, you can type in your query using the Raw Query Editor.

If using the Raw Query Editor, you should be familiar with syntax and operators used in Find All searches, since they are the same for defining raw queries. For instance, the following characters are used for the operators: And, Or, And Not, and Has Not.

| Operator | Character |
|----------|----------------------|
| And | Ampersand (&) |
| Or | Pipe symbol () |
| AndNot | Caret symbol (^) |
| Has Not | Exclamation mark (!) |

Parentheses are used to specify evaluation order. For instance, you want to search for stories that mention either Hillary or Bill Clinton, but you only want stories that do not also mention their daughter, Chelsea.

You could enter: `(Hillary & Clinton) | (Bill & Clinton) &!(Chelsea)`

The combination `&!` accomplishes the same thing as the caret symbol (`^`) would in a query. Another example is a query that searches for stories about domestic policies or domestic agenda, but not domestic abuse. In this example the query might be written this way:

`(domestic & agenda) | (domestic & policy) ^ (domestic & abuse)`

An asterisk (`*`) may be used as a wildcard for partial spellings, such as `airp*` to find stories with words such as airport, airplane, and airplay.

6. Use the Date Search area to define a time frame for the query; you can search by the date a story aired or was created, or by last date a story was modified.
7. (Optional) Maximum hits can also be altered in the Search Queue Setup dialog box. The default is 10. When a user opens a search queue, the search's result count is displayed in the status bar.
8. Click Save to save your new search queue. It will appear in the Directory panel.

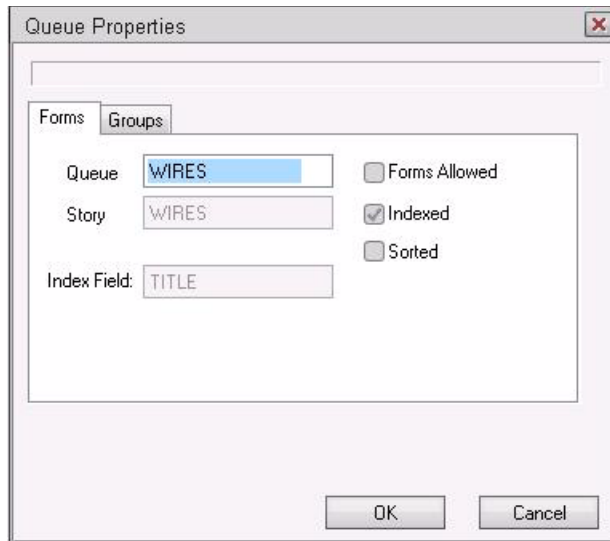


Search queues are distinguished from other types of queues by the icon. For search queues, a magnifying glass icon appears, as shown at left.

9. After the search queue is created, by default it will use the queue form and read group already associated with the parent directory in which the search queue was created. You can view these properties or modify them by doing the following:
 - a. Right-click on the new search queue in the Directory panel.
 - b. Select Properties.

An abbreviated version of the Queue Properties dialog box opens, showing only the Forms and Groups tabs.

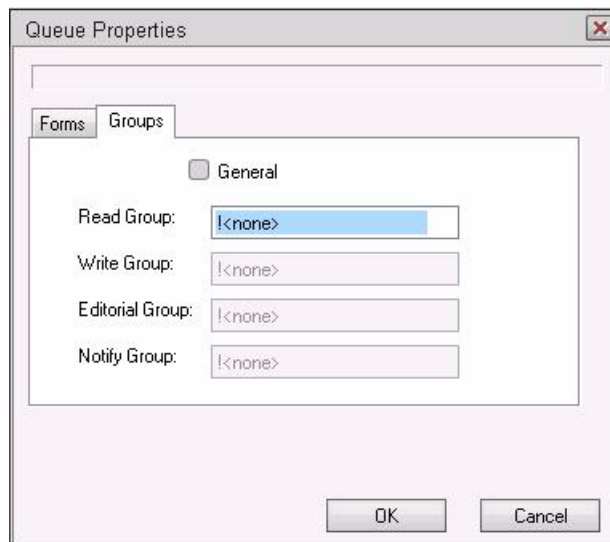
5 The Database: Directories, Queues, and Stories



The screenshot shows the 'Queue Properties' dialog box with the 'Forms' tab selected. The 'Queue' field is set to 'WIRES', the 'Story' field is 'WIRES', and the 'Index Field' is 'TITLE'. The 'Forms Allowed' checkbox is unchecked, 'Indexed' is checked, and 'Sorted' is unchecked. 'OK' and 'Cancel' buttons are at the bottom.

Only the queue form property on the Forms tab may be altered for a search queue. Though the other properties are displayed, they are read-only and not used by the system.

On the Groups tab, only the read group property may be altered. A setting of !<none> means no read group is defined; therefore, there is no restriction, and every user can see the search queue.



The screenshot shows the 'Queue Properties' dialog box with the 'Groups' tab selected. The 'General' checkbox is unchecked. The 'Read Group' field is set to '!<none>', while 'Write Group', 'Editorial Group', and 'Notify Group' are all set to '<none>'. 'OK' and 'Cancel' buttons are at the bottom.

- c. Click OK. If any settings were modified, they will be saved. To close without saving, click Cancel.



For more information about groups or forms, see “Groups” on page 151 or “Forms” on page 197. See also “Database Traits Summary” on page 122.

Viewing Search Queue Information from the Console

A system administrator can view some information about search queues from the console, by using variations of the list command.

To view information about a search queue:

- ▶ Use the following format: `list sq -v <search queue name>`

For instance:

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

For more information, see “list sq” on page 509.

Creating a New Story

In addition to opening existing stories, you can create new stories in a queue.

To create a story:

1. Open the queue in which you want to create a story.
2. Position your cursor in the queue below where you want the new story created.
3. Do one of the following:
 - ▶ Select File > New Story.
 - ▶ Press Ctrl+N.
 - ▶ Press the Insert key.

A new queue entry appears in the Queue panel as a blank row, and a blank story appears in the Story panel.

4. In the Slug column of the Queue panel, enter the new story’s title.
5. Enter any other important information in the remaining columns for the new story.
6. In the Story panel, enter the story’s text.

Using Script Templates

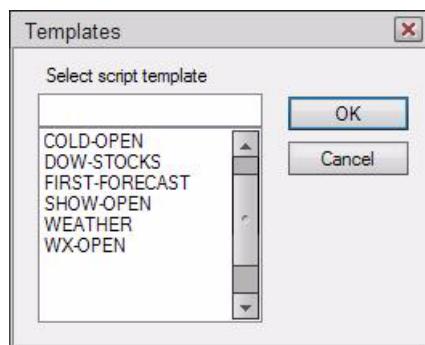
Script templates are templates created by an administrator for users to quickly insert predefined text into stories. The database structure for script templates is the same as the structure used for forms and lists:

```
SYSTEM.SCRIPT-TEMPLATES.<first letter>.<name>
```

For example, if a standard weather story has a set of production cues and presenter text that are always the same, you can create a story that will be a script template in SYSTEM.SCRIPT-TEMPLATES.W.WEATHER. In this story, place the standard text, presenter text, and production cues found in standard weather stories. After the template story is saved, the template is available to users via the Story menu or through an option in the Story Text sub-panel's context menu.

To use a script template:

1. Right-click in the Story Text sub-panel.
2. Select Insert Script Template. The Template dialog box appears with a list of all available templates.



3. Enter the name or select the template from the list.
4. Click OK.

After the template is chosen, the template is inserted in the story's body at the cursor position.



Script templates can be used to replace the use of complex macros to insert reusable material in stories.

Renaming a Directory or Queue

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



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

To rename a directory or queue in the database:

1. At the console, enter superuser mode.
2. Select *all* computers.
3. Take the system offline by typing the command: **offline**.
4. Broadcast a message instructing users to log out.
5. Log out all users on the system before renaming a queue or directory. This ensures that no stories are open for editing.



If users are not logged out, changes to stories may not be saved after the queue or directory is renamed. It is often most efficient to make several name changes at once. See “Logging Out All Users” on page 97 for more information.

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

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

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

```
NRCS-B# rename people staff
```

A message similar to the following appears:

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

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

```
NRCS-B# rename -v people staff
```



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

9. To continue with the renaming, type: `y`. A message similar to the following appears:

```
56 records will be modified [y/n]: y
```

```
Adding new directories...
```

```
Updating directory names...
```

```
56 directories renamed
```

```
1 directories added
```

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

```
TO: name too long
```



Before modifying files, the system checks for name length overflow. If any of the directory/queue names exceed their maximum length, no changes are made.

10. A verification request appears:

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

The user file is where the names of users' mail queues, home folders, and automatic destinations are stored. If you answer `y`, any item affected by renaming is changed automatically. If you answer `no`, you must change them yourself. Typically, answer `yes`.

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

```
23 user records modified
```

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



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

These references can include:

- Command bar icons set up by users
- Your system's service table
- Dialogs
- Keyboard description stories
- Server or Rx/Tx link job lists
- Wire distribution or keyword stories

- Your system's queue dictionary (/site/dict/queues)
12. Select *all* servers.
 13. Restart all devices by typing: **restart all**. You will see Hot-to-go messages as each device starts.
 14. Bring the system back online by typing: **online**. This allows users to log in.
 15. Press Ctrl+D to leave superuser mode.

Deleting a Directory or Queue

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

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



If a directory contains sub-folders or queues when you attempt to delete it, iNEWS will prompt you for confirmation. If you affirm the deletion, the directory and all its contents will be deleted from the system. Caution should be taken so that sub-folders and sub-queues are not inadvertently deleted.

To delete a directory or queue from the database:

1. Log in as a system administrator unless you have write access to the queue or directory.
2. Select the directory or queue you want to delete.
3. Do one of the following:
 - ▶ Select Tools > Delete Folder (or Delete Queue).
 - ▶ Right-click on the directory or queue, then select Delete Folder (or Delete Queue) from the context menu.

Recovering a Killed Story

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

To retrieve a story from the Dead queue:

1. Log in as a system administrator—that is, with a superuser account. This ensures you access to the Dead queue. On some systems, access to the Dead queue is restricted, and stories within it are set to retain their original read and write permissions.
2. Navigate to the Dead queue in the Directory panel and open it by double-clicking on it.

5 The Database: Directories, Queues, and Stories

3. Locate the story you want to recover in the Dead queue by scanning the list of stories displayed in the Queue panel for the story title (slug) or using the Find or Find All command. The Dead queue cannot be indexed, so you cannot use the Fast Text Search feature.
4. Select the story or stories you want to retrieve by doing one of the following:
 - ▶ Click on the selector button located to the left of the story’s row in the Queue panel. The entire row is highlighted when selected.
 - ▶ Move the cursor to the row and press Shift-Spacebar.
 - ▶ Click on each row’s selector button while holding down the Control (Ctrl) key to select multiple stories. The Shift key can be held down if you want to select all story rows between two mouse clicks.
5. Copy the selected story (or stories) to the new location by doing one of the following:
 - ▶ Click on and drag the highlighted selection to another queue in the Directory panel and release. The selected stories will be copied to the new queue location.
 - ▶ Use the Copy and Paste buttons or the Edit menu options to copy and paste the highlighted selection into a new queue location.
 - ▶ Use the Duplicate command to copy the highlighted selection to another location—particularly if the Dead queue is read-only.
 - ▶ Drag the story or stories from the Dead queue into another queue in another workspace. This lets you position the story or stories where you want them.

Viewing Database Traits

You can get information about your iNEWS database from both the iNEWS Workstation and the console. Which one you use depends on what information you want to view.

| Information Wanted | Source to Use |
|--|-------------------|
| Information about a directory or queue | iNEWS Workstation |
| Information about several directories or queues simultaneously | iNEWS Console |
| Information about stories in the database | iNEWS Console |

This section provides procedures for viewing database traits for directories or queues from either the workstation or the console. For more information about getting information on stories, see [“Viewing Information about Stories” on page 118](#).

To view database information about directories or queues from the console:

- ▶ Use the following format of the list command: `list d <folder/queue name>`

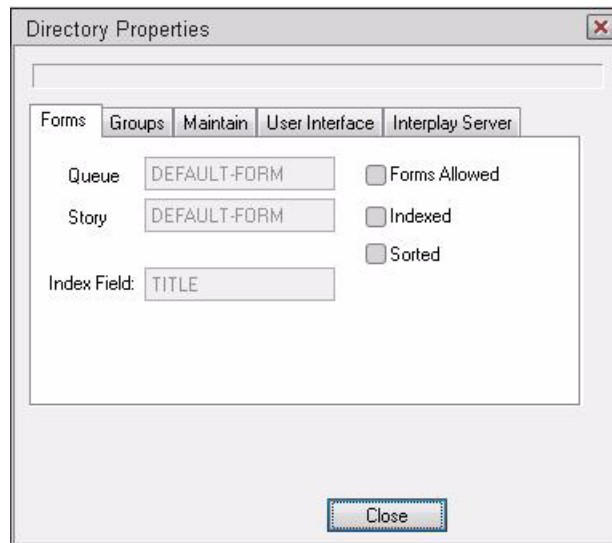


This command has a verbose option, `list d-v`, which gives you more detailed information.

To view database information on a specific directory or queue from an iNEWS Workstation:

1. Log in to iNEWS at a workstation.
2. Open an iNEWS Workspace.
3. Navigate to the directory or queue you want in the Directory panel.
4. Right-click on the name of the directory or queue.
5. Select Properties in the context menu.

The Directory/Queue Properties dialog box shows you the properties (traits) for the directory or queue you selected; however, its look can vary. For instance, the Locks tab does not appear when viewing properties of directories. If you are not logged in as a system administrator, and no database manager account was created in iNEWS, the dialog will appear like this:



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

5 The Database: Directories, Queues, and Stories

If a database manager account exists, then a Database manager login button will appear in the bottom left corner of the dialog box. See “[Changing Database Traits](#)” on page 136 for more information.



For more information about the various database traits available from the iNEWS Workstation, see “[Database Traits Summary](#)” on page 122.

Viewing Information about Stories

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

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



The list q command has a verbose option that gives you more detailed information. For instance, a verbose list, such as list q-v, includes read and write group information for each story in the queue. Read and write groups are explained in “[Groups](#)” on page 151. Also, see “[Managing Traits at the Console](#)” on page 623.

The *<record limit>* specifies the number of stories from the queue you want to list. Queues might contain thousands of stories so a command without a specified *<record limit>* might scroll large amounts of output. For instance, to limit the list to the first three stories in PEOPLE.CAROLYN.NOTES, type: `list q people.carolyn.notes 2`

A message similar to the following appears:

```
PEOPLE.CAROLYN.NOTES.SEARCH id=449889

rec quick index  LHDM-WObfpRmF  f.id  time  modified-time
2   pm-chronology --DM-W-----  457243 165   Jul 10 16:16:39 2010
3   pm-thumbnails --DM-W-----  487595 163   Jul 10 16:21:17 2010
```

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

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

| Flag | Status Information | Flag | Status Information |
|------|--------------------|------|---|
| L | Locked | b | Story’s body (text) is edit locked. |
| H | Held | f | Story’s fields are edit locked. |
| D | Duplicated | p | Story’s production cues are edit locked |

| Flag | Status Information | Flag | Status Information |
|------|--------------------|------|--------------------|
| M | Modified | R | Read only |
| - | ----- | m | Mail |
| W | Wire | F | Floating |
| O | Ordered | | |

You cannot change any of these flags from the console, except the edit-locked status, which you can remove from a story with the `unbusy` console command. For instructions, see [“Unbusy Stories and Queues” on page 147](#).

To list information for a particular story:

- ▶ Use the following format of the list command:
`list qindex=<index value> q <queue name>`

The *index value* is the value of the selected sort field of the story you want to list. This value is typically the text found in the title field, but you can set different fields as the index field.

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

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

A screen similar to the following appears:

```
PEOPLE.SMITH.NOTES id=449889
rec quick index LHDM-WObfpRmF f.id time modified-time
3901 bc-exp--nomad --DM----- 420690 165 Jul 6 20:23:11 2010
```

In this example of the story in PEOPLE.SMITH.NOTES, the D and M flags appear, indicating the story is duplicated and has the modified flag set.

The quick index value can be uppercase or lowercase and must be a single word with no spaces unless you precede the space with a backslash (\) or put the index value inside quotation marks (“ ”), such as either of these examples:

```
list qindex=terror\ suspect q show.10pm.rundown
```

```
list qindex="terror suspect" q show.10pm.rundown
```

Viewing Who Moved, Duplicated, or Killed a Story

Additional options for the list command can also tell you the last person to move, duplicate, or kill a story (sending it to the Dead queue).

To list the last person to move, duplicate, or kill a particular story:

- ▶ Use the following format of the list command:

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

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

The *search word* is a word from the Index (sort) field of the story. It is not case-sensitive. It must be a single word with no spaces unless you precede the space with a backslash (\) or put the search words inside quotation marks (" "). The *record limit* is the numerical limit of stories provided in response to your list command—for instance, the most recent stories killed in the Dead queue.

To list information for each story in a queue without using a word in the Index field:

- ▶ Use the following format of the list command:

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

The b in the command is optional and stands for backwards.

- list q-m — lists stories beginning with the oldest story in the queue
- list q-mb — reverses the order and lists stories from the most recent material in the queue, such as the most recently killed stories in Dead.

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

For instance, type: **list q-b dead 5**

A screen similar to the following appears:

```
DEAD id=123231
rec quick index      LHDM-WObfpRmF f.id    time  modified-time
0                ---M--O----- 314490  0      Sep 6 09:51:58 2010
-1  kyw-directors  --DM--O----- 313587  15     Sep 5 11:47:33 2010
-2  a              ---M----- 161746  11     Sep 6 09:15:06 2010
-3  sep 2010       ---M----- 314093  0      Sep 5 17:34:19 2010
-4  008            --DM--O----- 313555  600    Sep 5 16:49:24 2010
```

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

```
list q-mb dead 5
```


A screen similar to the following appears:

```
DEAD id=123231
rec quick index LHDM-WObfpRmF f.id time user name
0          ---M--O----- 314490 0 palmer
-1 kyw-directors --DM--O----- 313587 15 williams
-2 a          ---M----- 161746 11 adbpurge
-3 sep 2000   ---M----- 314093 0
-4 008        --DM--O----- 313555 600 ragusa
```

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

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

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

A screen similar to the following appears:

```
ASSIGNMENTS.MONDAY id=14569
rec quick index LHDM-WObfpRmF f.id time user name
0 camera        --DM----- 16217 274 williams
```

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



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

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

```
NRCS-A# list q-mb dead 5000 | doc -ptu people.p.palmer.notes
```

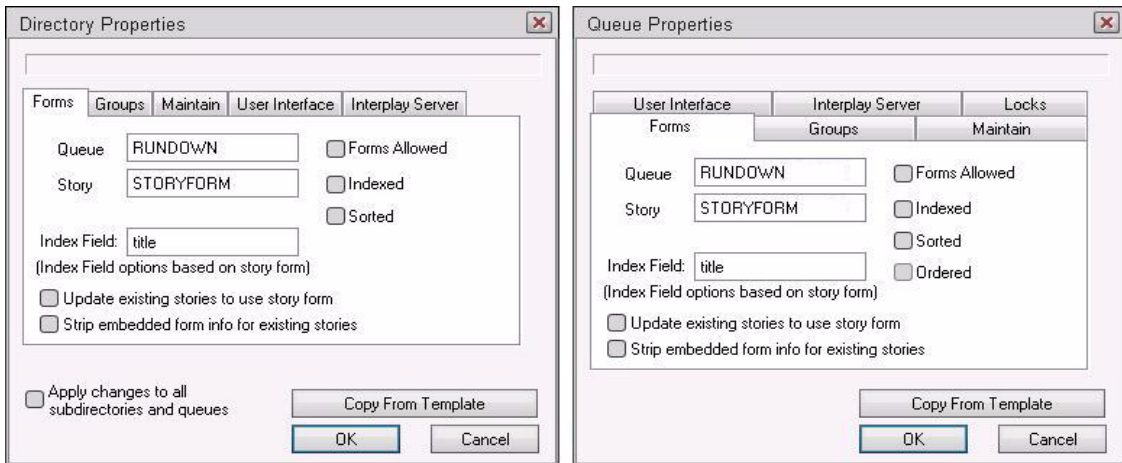
Database Traits Summary

Assigning traits can be done from the iNEWS Workstation as well as the console. For information on viewing and altering database traits from the console, see Appendix G.

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

Directory/Queue Properties Dialog Box

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



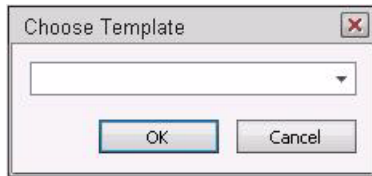
Other differences include check boxes and tabs. For instance, the Apply changes... check box—shown in graphic on the left—only appears in the bottom left corner of the Directory Properties dialog box. Also, the Ordered check box—shown in graphic on the right—only appears in the right column of the Forms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

Both, however, have a Copy From Template button, which can be used to apply directory or queue property settings from a template to the chosen folder or queue, respectively. This can be very helpful when adding a new newscast to the database file structure.



Avid recommends the use of templates to help speed up the process of future configuration modifications as well as to help avoid the potential for user error.

When you click the Copy From Template button, the following dialog box opens with a drop-down list of available templates from which to choose.



The templates in the list are created as queues in SYSTEM.PROPERTIES.

The number of tabs in the Directory/Queue Properties dialog box varies, depending on whether you choose to view properties for a queue or a directory. The tabs are:

- Forms
- Groups
- Maintain
- User Interface
- Interplay Server
- Locks (This tab only appears in the Queue Properties dialog box.)



Each tab is explained in the following sections.

Forms Tab

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



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

| Trait Options | Description |
|---|---|
| Queue | <p>The Queue drop-down list lets you select a form to display information in the Queue panel. The form defines what fields appear, which should be a sub-set of the fields used in the story form. A field included in the queue form that does not actually exist in the story form cannot be written to in the Queue panel. When !<none> is selected, no form is applied. This drop-down list is the equivalent of the qform database trait (dbtrait) at the console.</p> |
| Story | <p>The Story drop-down list lets you select a form to display information in the Story Form panel. When !<none> is selected, no form is applied. This drop-down list is the equivalent of the sform dbtrait at the console.</p> |
| Index Field | <p>The Index Field drop-down list lets you select a form field that will be used to sort the queue alphabetically. The cursor is placed on this form field by default when a user displays stories in a queue. This drop-down list is the equivalent of the sortfield dbtrait at the console. For more information, see “Index Field/Story Form Compatibility Error Messages” on page 126.</p> |
|  | <p><i>The optional fields in the Index Field drop-down list depend on the form selected in the Story drop-down list on the Forms tab. For instance, if you select a story form that only contains two fields, such as Title and Writer, then those two fields will be the only options listed in the Index Field drop-down list.</i></p> |
| Update existing... | <p>The Update existing stories to use story form check box, when selected, changes the story form assignment for previously existing stories within a queue. This check box is the equivalent of the cform dbtrait at the console.</p> |
| Strip embedded... | <p>The Strip embedded form info for existing stories check box, when selected, removes embedded form traits from stories. For instance, queues with the Forms Allowed trait stamp the look of the story form into the story. Assigning a different story form to one of these queues and selecting Update existing... check box will not affect the look of stories with the embedded forms. You would need to strip the embedded “look” from the story so it would then use the form assigned to the queue it is in. This check box is the equivalent of the stripform dbtrait at the console.</p> |
|  | <p><i>The Update existing... and Strip embedded... check boxes are not database traits, but rather, they are used to apply current settings and/or changes in the dialog box at present. This means they will always appear unchecked when the dialog box opens, and they will not appear at all if the Directory/Queue Properties dialog box is opened in read-only mode.</i></p> |

| Trait Options | Description |
|---------------|--|
| Forms Allowed | <p>The Forms Allowed check box must be assigned to all queues in the Forms directory. The forms will not work without this database trait applied. Additionally, this trait can be assigned to any queue in the database, but is usually only assigned to other queues that receive stories from other systems via rxnet/txnet and then build forms for those stories, as needed. This check box is the equivalent of the dbtrait +f -f at the console.</p> |
| Indexed | <p>The Indexed check box, when selected, applies the Index trait. This trait is assigned to queues and directories you want to be indexed by the Fast Text Search (FTS) server. This allows for quicker searching of the queue or directory.</p> <p>This check box is the equivalent of the dbtrait +index -index at the console. See “Batch Indexing” on page 377 for more information.</p> |
| Sorted | <p>The Sorted check box, when selected, applies the sort trait, which determines whether or not the stories in a queue will be sorted. Queues with the sort trait are sorted by the form field you choose in the Index Field drop-down list. For instance, you may want to sort a rundown queue by the Page Number field, so when a user changes the numbering in the fields of that column, the rows automatically adjust to the numerical order. See “Turning Off the Ordered Trait of a Sorted Queue” on page 138 for more information. This check box is the equivalent of the dbtrait +so -so at the console.</p> |
| Ordered | <p>The Ordered check box only appears on the Queue Properties dialog box, not the Directory Properties dialog box. The Ordered check box is a unique check box, because it might appear as read-only, depending on the circumstance. It is provided to show you whether a queue is currently ordered. This is particularly helpful in identifying queues that have the sort trait, but are no longer being sorted because a user manually adjusted the order of stories in the queue. If a sorted queue was manually ordered, the check box appears white and contains a checkmark, which you can remove if you want to reinstate the sorting feature. Unchecking the check box is the equivalent of the dbtrait, -o, at the console. See “Turning Off the Ordered Trait of a Sorted Queue” on page 138 for more information. However, if a queue is not ordered at present, then the box will appear gray and empty. You cannot select this box to order a queue.</p> |

Index Field/Story Form Compatibility Error Messages

A *story form* is a form that defines the fields displayed in the Story Form panel of the iNEWS Workspace. The fields typically consist of important information about the stories kept in the queue, such as the title, writer's name, and the dates the story was created or modified. The field chosen as the Index Field in the Queue Properties dialog box serves two purposes:

- It is the field used to sort stories in sorted queues.
- When the Index Field is selected during a Find or Find All search of a non-indexed queue, the system is able to return search results faster.

The Index Field is not associated with Fast Text Searches (FTS).

Therefore, it is crucial that the field selected as the Index Field actually exists as part of the story form. For this reason, the system will check for compatibility between the settings of the Story drop-down list and the Index Field drop-down list. Depending on the list selections, various messages may appear.

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

However, if the user changes the story form from Rundown to another form that does not have the Audio-Time field, the following message appears.



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



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

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

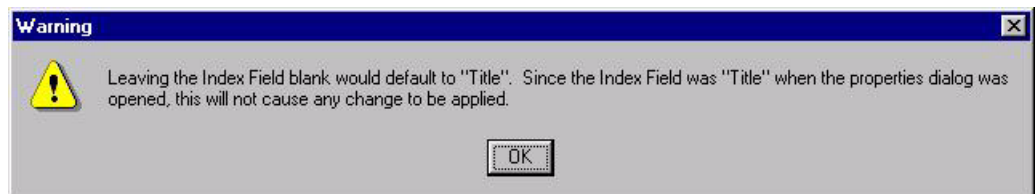


This means the *Index Field* drop-down list will be inaccessible and appear grayed out. When the index field setting is blank, the system will use the default field, which is the *Title* field (also called the *Slug* field).

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



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

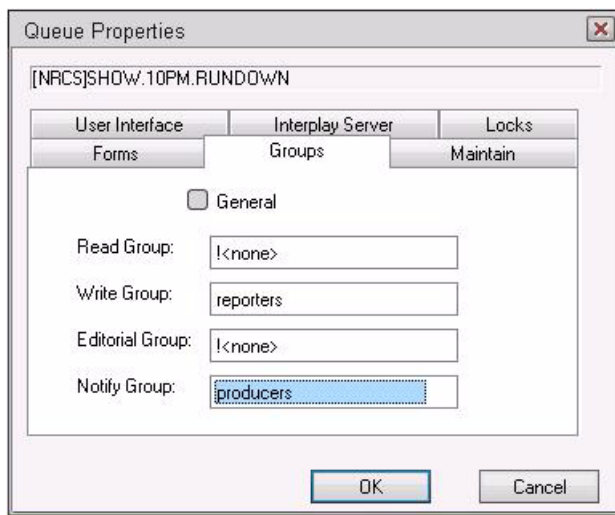


5 The Database: Directories, Queues, and Stories

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

Groups Tab

This section provides information about database traits available as options on the Groups tab of the Queue Properties dialog box.



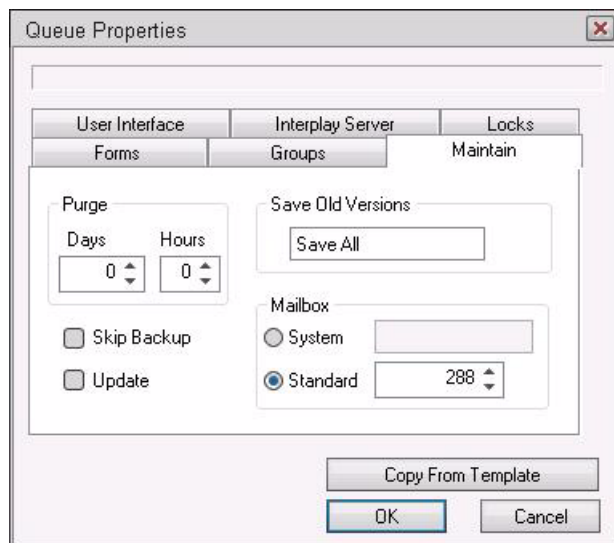
See “Groups” on page 151 for more information about groups.

| Trait Options | Description |
|---------------|---|
| General | <p>The General check box, when selected, specifies that stories moved to the queue will retain their original security restrictions, namely, their read and write groups. For instance, in some cases, sites restrict access to the Dead queue, but to further guarantee security of stories, the General trait is applied to the Dead queue. This prevents users who do have access to the Dead queue from opening stories they could not have opened in their original queues.</p> <p>This check box is the equivalent of the dbtrait +g -g at the console.</p> |

| Trait Options | Description |
|-----------------|--|
| Read Group | <p>The Read Group drop-down list lets you restrict read access to a queue or directory to a group of users. Users who are not in the read group cannot see the directory or queue. When !<none> is selected, no group is applied; therefore, all users will have read access to the queue or directory. The groups are not created here.</p> <p>This drop-down list is the equivalent of the dbtrait readgroup or rg at the console.</p> |
| Write Group | <p>The Write Group drop-down list lets you restrict write access to a queue or directory. Users who are not in the write group cannot add or modify data in the directory or queue. When !<none> is selected, no group is applied; therefore, all users will have write access to the queue or directory. The groups are not created here.</p> <p>This drop-down list is the equivalent of the dbtrait writegroup or wg at the console.</p> |
| Notify Group | <p>The Notify Group drop-down list lets you specify what group of users is notified whenever stories are added to or modified in a queue. When !<none> is selected, no group is applied; therefore, no users will be notified of additions or modifications to the queue or directory. The groups are not created here.</p> <p>This drop-down list is the equivalent of the dbtrait notify or ng at the console.</p> |
| Editorial Group | <p>The Editorial Group drop-down list lets you restrict editorial access to a queue or directory. Users who are not in the editorial group cannot add breaks, float stories, or reorder queues, nor can they delete, remove or move data from a queue or directory. When !<none> is selected, no group is applied; therefore, only a user who has Write privileges to the queue can add breaks or float, reorder, delete, or move stories from the queue.</p> <p>This drop-down list is the equivalent of the dbtrait editorialgroup or eg at the console.</p> |

Maintain Tab


This section provides information about database traits available as options on the Maintain tab of the Queue Properties dialog box.



| Trait Options | Description |
|-------------------|--|
| Save Old Versions | <p>The Save Old Versions drop-down list determines how many old story versions are retained in each queue. The Save Old Versions drop-down list is the equivalent of the dbtrait <code>save-n o a</code> at the console. Options include:</p> <ul style="list-style-type: none"> • Save None – Retains no old versions of a story when a new version is saved in the queue. • Save Previous – Retains the previous version of a story when a new version is saved in the queue. • Save Original – Retains the original version of a story when a new version is saved in the queue. • Save All – Retains all versions of a story when a new version is saved in the queue. |

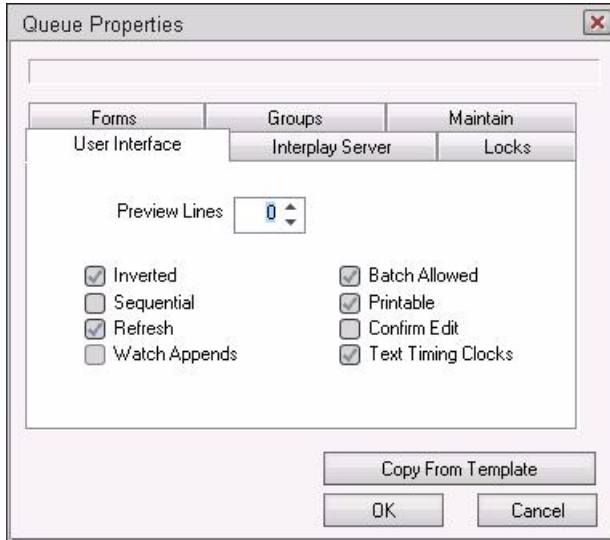


The Save Old Versions trait is queue-specific. For instance, a story is moved from a queue that saves all versions to a queue saving none. In this case, all versions are moved to that queue, but the next time the story is edited and saved, the old versions are sent to the Dead queue. The command `dbdump` honors the Save Old Versions when set to Save None and will only dump the latest version of a story in the queue with that trait.


| Trait Options | Description |
|---|--|
| Skip Backup | The Skip Backup check box determines whether or not a directory or queue is left out of database backups. This check box is the equivalent of the dbtrait +x -x at the console, and is also known as a skip flag. |
| Update | The Update check box indicates whether or not the stories in a queue will be replaced as new versions are moved or copied to it. This check box is the equivalent of the dbtrait +u -u at the console. |
|  | <i>The Update trait does not affect stories restored from tape backups. If you restore a story to a queue that already contains a version of that story, you will have two versions of the same story, even if the queue is assigned the update trait.</i> |
| Mailbox section | The Mailbox section does not apply to the e-mail feature of iNEWS. These mailboxes are “signal carriers” by which utility programs, called servers, are notified to perform a pre-defined task. See “Mailbox Tasks” on page 325 for more information. |
| | This section’s trait is the equivalent of the mail dbtrait at the console. |
| | There are two types of mailboxes: |
| | <ul style="list-style-type: none"> • System – The System radio button and drop-down list are used to assign mailboxes reserved for system functions, such as the keyboard and form checkers. Each queue or directory that needs a reserved system mailbox is assigned the correct one when the system is installed by Avid Customer Support personnel. Options include: All, Keyboard, Keyword, Distribution, and Group. See “Reserved Mailboxes” on page 327 for more information. • Standard – The Standard radio button and spin box are used to assign mailboxes to queues. These are mailboxes used by utility programs you can configure, such as action servers. The mailbox number assigned to the queue must match the mailbox number of the server monitoring it, as defined in the /site/config file. Valid mailbox numbers are one through 5000. |
| Purge | The Purge section allows you to set the reoccurrence schedule for purging a queue. The purge interval determines how old stories in a queue can get before they are purged. Every hour, your system removes any stories that are older than their queue’s purge interval and places the stories in the Dead queue. For more information, see “Database Purge Intervals and Limits” on page 139 . |
| | The Purge section’s Days and Hours spin boxes are the equivalent of the purgeinterval dbtrait at the console. |

User Interface Tab

This section provides information about database traits available as options on the User Interface tab of the Queue Properties dialog box.



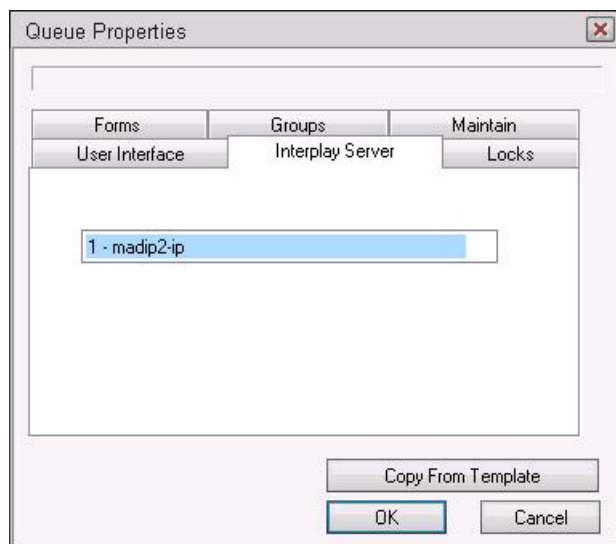
| Trait Options | Description |
|---------------|---|
| Preview Lines | <p>The Preview Lines spin box allows you to set a number of lines per story that will appear as a preview in the Queue panel. Usually, a queue will only show one line of information per story, similar to what appears in the fields of the Story Form panel. By applying the preview trait, users can also see a preview of each story’s text in the Queue panel, without having to open the entire story. A setting of zero will show the one line of information that is the standard; A setting of one will show that line plus one line of text, and so forth. This trait can be overridden by a user’s preferences. See “Layout Tab” on page 80 for more information.</p> |
| | <p>The maximum number of preview lines allowed is 22. This spin box is the equivalent of the dbtrait, dis, at the console.</p> |
| Inverted | <p>The Inverted check box, when selected, will force the most recent stories in a queue to be displayed at the top. Otherwise, the most recent stories will appear at the bottom.</p> |
| | <p>This check box is the equivalent of the dbtrait +i -i at the console.</p> |

| Trait Options | Description |
|---|--|
| Sequential | <p>The Sequential check box, when selected, will force a directory to list its contents—of sub-directories and queues—in the order in which they were created. Otherwise, the contents are listed in alphabetical order.</p> <p>This check box is the equivalent of the dbtrait +s -s at the console.</p> |
| Refresh | <p>The Refresh check box, when selected, assigns the Refresh trait to a queue, so the system will begin automatically refreshing your screen when changes are made in the queue. This means when you are looking at a queue in the Queue panel, you will immediately see changes made to that queue by other users.</p> <p>This check box is the equivalent of the dbtrait +refresh -refresh at the console.</p> |
|  | <p><i>Use the Refresh trait only on important queues, like rundown queues that are often modified by multiple users simultaneously. To automatically refresh a queue, your system must spend a lot of time monitoring workstations where users are viewing that queue. Assigning the refresh trait to too many queues that are often accessed at the same time greatly increases the amount of work your system has to do and may severely degrade its overall performance.</i></p> |
| Watch Appends | <p>The Watch Appends check box, when selected, lets a queue monitor incoming data for new stories sent by the wire service, appends them to the wire queue, and immediately displays them to users who have that wire queue open. While this trait can be applied to any queue, it is crucial that it be assigned to queues that receive wire service data, such as the WIRES.ALL queue.</p> <p>This check box is the equivalent of the dbtrait +w -w at the console.</p> |
| Batch Allowed | <p>The Batch Allowed check box, when selected, indicates whether or not the kill, move, or duplicate operations can be performed against an entire queue. Unchecking this box 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>This check box is the equivalent of the dbtrait +q -q at the console. See “Managing Traits at the Console” on page 623 for more information on Batch Allowed, also known as Queue Operations Allowed.</p> |
| Printable | <p>The Printable check box indicates whether you can use the print command to print all stories in the queue with a single command. This trait is usually applied to rundown queues, such as SHOWS.5PM.RUNDOWN. This trait does not limit or prevent the ability to print a single story (or script) in a queue.</p> <p>This check box is the equivalent of the dbtrait +p -p at the console.</p> |

| Trait Options | Description |
|---------------|---|
| Confirm Edit | <p>The Confirm Edit check box, when selected, will instruct iNEWS to prompt for confirmation before letting a user edit a story. This trait is typically assigned to queues in which users are likely to read but not change stories. It should not be assigned to queues with stories that are edited often. Doing so will needlessly slow down your users.</p> <p>This check box is the equivalent of the dbtrait +r -r at the console.</p> |
| Text Timing | <p>The Text Timing check box, when selected, activates the timing clocks that appear at the bottom of the Story panel for all stories in the queue. The clocks are:</p> <ul style="list-style-type: none">• TTC – Time from start of story to cursor• BLK – Time of blocked (or highlighted) text• EST – Estimated read time of entire story <p>This check box is the equivalent of the dbtrait +t -t at the console.</p> |

Interplay Server Tab

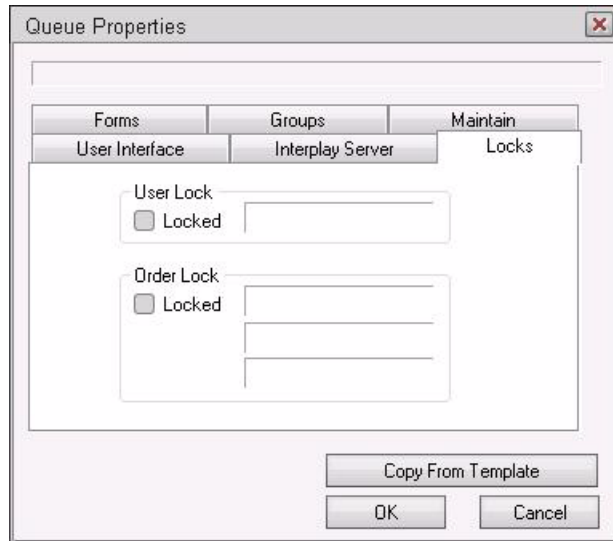
This section provides information about database traits available as options on the Locks tab of the Queue Properties dialog box.



The Interplay Server tab lets you select the server that will integrate with iNEWS. When configured in SYSTEM.INTERPLAY, a list of up to ten options will appear as a list.

Locks Tab

This section provides information about database traits available as options on the Locks tab of the Queue Properties dialog box.



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

| Trait Options | Description |
|---------------|--|
| User Lock | The Locked check box in the User Lock section indicates that a user has either acquired an easy lock or a key lock on the queue. The User Lock field indicates the name of that user. If the queue is easy locked, access to the locked queue is limited to a system administrator (with a superuser account) or the user who locked the queue. If the queue is key locked, access to the locked queue is limited to a system administrator (with a superuser account) or any user who knows the key (password). The check box and field depict the current condition of the queue, so both are blank when the queue is not locked. However, you can find out the name of the last user to have locked the queue by going to the console. For more information, see “Identifying Locked Queues and Stories” on page 141. |

| Trait Options | Description |
|---------------|--|
| Order Lock | The Locked check box in the Order Lock section indicates the queue is order locked at present, which limits who can rearrange the order of stories in the queue. The Order Lock fields indicates the user name and device which implemented the order lock, and when it happened. The check box and fields depict the current condition of the queue, so both are blank when the queue is not order locked. However, you can find out the name of the last user to have order locked the queue by going to the console. For more information, see “Identifying Locked Queues and Stories” on page 141. |

Changing Database Traits

You must be logged on at an iNEWS Workstation as a system administrator—that is using an account with the superuser trait—or provide the database manager (dbmanager) password to change database traits of directories and queues. For more information on dbmanager, see [“The Database Manager Account” on page 97.](#)

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

When you change a directory’s traits from a workstation, the changes only affect the directory you selected, unless you specify otherwise. This is directly opposite to what happens when changing database traits at the console.

This section provides the procedure for changing traits from a workstation. For more information on how to change traits from the console, see [“Managing Traits at the Console” on page 623.](#)

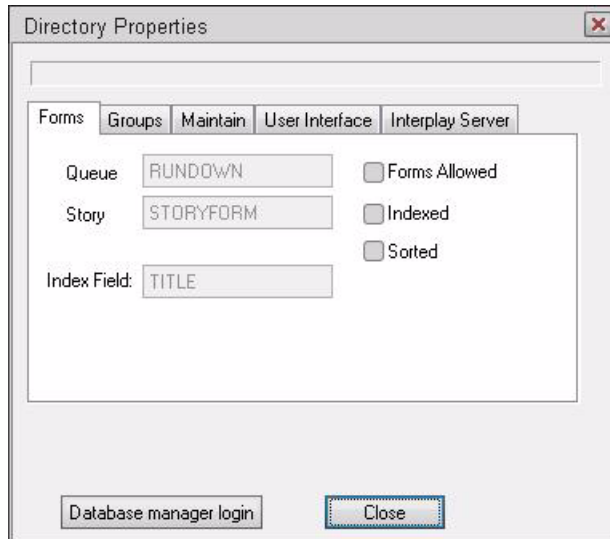


Avid recommends you ensure that no users are working in a directory or queue prior to altering the database traits of that directory or queue.

To change database traits from a workstation:

1. Navigate to the directory or queue you want to change in the Directory panel.
2. Right-click on it.
3. Select Properties from the context menu. One of two things will happen, which will determine what you are to do next.
 - ▶ If you are logged on as a system administrator, the Directory/Queue Properties dialog box will appear. Go to step 5.

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



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



The Database Manager Password dialog box closes, and the Database manager login button disappears, and the dialog box with a check box.

Go on to step 5.

5. Do one of the following:
- ▶ If altering traits of a queue, go to step 6.
 - ▶ If altering traits of a directory, and if you want the changes you make next to apply to all queues and subdirectories in the parent directory, click on the check box labeled Apply changes to all subdirectories and queues.

5 The Database: Directories, Queues, and Stories

- ▶ If altering traits of a directory, but you only want to change the traits of the directory chosen when the Directory Properties dialog box opened, do not select the Apply changes ... check box.



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

6. Make changes to the various traits as needed. These traits are explained in detail in [“Database Traits Summary”](#) on page 122.




Selecting and/or unselecting check boxes in the Directory/Queue Properties dialog box does not apply changes immediately. Only clicking the OK button does that.

7. Click OK to save and apply changes.

Turning Off the Ordered Trait of a Sorted Queue


If someone manually rearranges stories in a sorted queue, the Ordered trait is applied to that queue and the sorting function is disabled even though the Sorted trait is still applied to the queue. You can remove the Ordered trait, which will restart the queue’s sort function from either the console or any iNEWS Workstation.

This section provides the procedure for doing it from a workstation. For more information on how to do this from the console, see [“Turning Off the Ordered Trait of a Sorted Queue”](#) on page 138. When a sorted queue is ordered, the Queue Properties dialog box will appear similar to this:

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

To remove the Ordered trait and allow the queue to resume its original sorting function:

1. In the Queue Properties dialog box, deselect the Ordered check box.
2. Click OK to save the change.

 *When removing the checkmark from the Ordered check box and clicking OK—the equivalent of `dbtraits -o`—the current contents of the queue will be sorted immediately. Any stories added to the queue or stories edited in the queue will be sorted.*

Database Purge Intervals and Limits

Purging is the first step in the process of freeing up space. Only when stories are removed from the Dead queue by the `dbserver` program is any database space freed.

Any queue to which the system automatically sends stories should be purged regularly. For instance, queues in the Wires directory should be purged often, or else the database fills up with old wire stories. If you set up a keyword story to send stories to a queue, ensure that queue is purged regularly. Otherwise, you risk running low on space.

5 The Database: Directories, Queues, and Stories

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

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

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

Considerations for Choosing Purge Intervals

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

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

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



Old stories restored from tape are treated as newly created stories unless `dbrestore -M` was used when the stories were restored from tape. The `-M` option preserves the modification time of the story recorded when the story was dumped to tape.

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

When choosing a purge interval, pay attention to queues likely to contain copies of stories held in other queues. Copies of stories are really pointers back to the original story. The actual story cannot be removed until all the copies have been purged. Otherwise, the copies would not have a story to which they could point.

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



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

Identifying Locked Queues and Stories

This section provides information on how to determine when a story or queue is locked and by whom. It also has a reference section that provides details about the types of locks in iNEWS.



As shown at left, a padlock appears over queue icons in the Directory panel so you can identify currently locked queues from an iNEWS Workstation.



A similar padlock—shown at left—appears on the selector buttons in the Queue panel when stories are locked.

While the padlocks do not tell you who initiated the lock, you can find that out for currently locked queues at the iNEWS Workstation. If a queue or story is not locked at present, but you want to know who last locked it, you must use the console. You can find out more from the console about locked queues and stories, even if they are no longer locked. The procedures for using the workstation or the console are covered in this section.

To find out who locked a currently locked queue from a workstation:

1. Right-click on the queue in the Directory panel.
2. Select Properties.
3. In the Queue Properties dialog box, select the Locks tab. The information is located on that tab.

To find out the last user to have locked a queue that is no longer locked:

- ▶ At the console, use the list command in the following format:

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

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

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

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

```
SRPlo-LIisUGQsXWFiT  lockuser  directory
Q-----I----A----   edmonds   PEOPLE.SNOW.MAIL
```

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



The L flag of the list command output shows whether the queue is locked at present. For more information about the letter flags of the list command, see “Managing Traits at the Console” on page 623.

To find out who last order locked a queue:

- ▶ At the console, use the list command in the following format:

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

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

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

The name of the person who last rearranged the order of the sorted queue appears in the lockuser column. The number of the device from which the order was initiated also appears.

```
SRPlo-LIisUGQsXWFiT  orderuser  device  directory
QSRP-----sU-Q-----  williams   301     PEOPLE.SNOW.MAIL
```

In this case, williams was the last person to have order locked the queue. If the queue has never been order locked, the orderuser name will be blank and the device number will be zero (0). If no queue name is entered in the command, the order information for all folders and queues is displayed.



For information on how to remove the order lock and resuming sorting in a sorted queue, see “Turning Off the Ordered Trait of a Sorted Queue” on page 138.

To find out who last locked a story:

- ▶ At the console, use the wholockedit command in the following format:

```
wholockedit [<queue name>]
```

For instance, to find out who locked each story in the SHOW.10PM.RUNDOWN queue, type: `wholockedit show.10pm.rundown`

A screen similar to the following appears:

```
rec(18) locked by danielmi
      modified by danielmi on 2010-02-09 11:53:20
page-number   A14
v-shot
presenter     K
title         MAYORAL RACE
v-format      TAG
v-graphics
v-audio
audio-time    0:15
runs-time     0:00
total-time    0:15
back-time
status        OK
endorse-by
modify-date   2010-02-09 11:53:20
modify-by     danielmi
v-edby
v-editbay
v-tapenumber
v-writer
video-id
item-channel
event-status
air-date
```

In the previous example, the story, MAYORAL RACE, is the only one in the queue that is locked. The last user to lock and the last user to modify the story are provided; they aren't always the same. If no stories are locked in the queue, the system will display the message: `no stories locked`.

To see the status of every story in the queue, add `all` to the end of the command, after the queue name.

Types of Locks

There are four types of locks that can apply to either stories or queues:

- Edit lock
- User lock
- Order lock
- Production lock

Not all of the locks apply to both stories and queues. All four types are explained in the following sections. For information on how to unlock stories or queues, see [“Removing Locks from a Workstation” on page 146](#) and [“Unbusy Stories and Queues” on page 147](#).

Edit Lock





An edit lock is applied to a story to prevent multiple users from editing the story at the same time. The edit lock is applied in two ways:

- Manually – when a user types Ctrl+E. This locks both the story form and body when neither are already locked by the system.
- Automatically – by the system, which occurs in a segmented way when a user begins typing in either the story body or the story form.

For instance, when a user types in the Story Text panel, the system locks the story’s body. But another user can still type in the Story Form panel, which results in an automatic lock of the story’s form for that other user.

When a user attempts to manually lock a story by typing Ctrl+E, the system checks for segmented edit locks. If one exists for either segment—the body or form—iNEWS notifies the user.

Additionally, edit lock indicators are provided on the Status bar.

| Indicator | Description |
|---|--|
|  | The user, danielmi, locked the story form. |
|  | The user, danielmi, locked the story’s body. |
|  | The user, danielmi, locked the entire story, body and form. |
|  | The user, jeff, locked the story body, but the user, danielmi, has the Edit lock for the story form. |

The edit lock is removed automatically when a user navigates to another location in the database. However, a user can manually remove the Edit lock while remaining in the story. For more information on how to do that, see [“Removing Locks from a Workstation” on page 146](#).

If a story is mistakenly left edit locked, it is considered to be in a busy state. You must unbusy the story before anyone can edit it. For more information, see [“Unbusy Stories and Queues” on page 147](#).

User Lock

A user lock is a lock that is manually (or purposefully) applied to a story or a queue by a user to limit who can see the contents of the queue or story. Additionally, there are two types of user locks:

- A Key lock is when a user locks the queue and applies a password, known as a key, to the queue. At that point, only system administrators and users who know the key can access the queue or story.
- An Easy lock is when a user locks the queue or story and applies their User ID as the key to the queue. At that point, only system administrators and users who log in using that User ID can access the queue or story. When the user forgets the key or is unavailable to log in and unlock the queue or story, you must remove the lock before anyone can access the queue or story. For more information, see [“Removing Locks from a Workstation” on page 146](#).

Order Lock

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

If a queue is mistakenly left order locked, it is considered to be in a busy state. You must unbusy the queue from the console before anyone can order it. For more information, see [“Unbusy Stories and Queues” on page 147](#).

Production Lock

A production lock is similar to an order lock in that it prevents multiple users from changing the order of a queue simultaneously. The difference is that a production lock is manually applied to a queue by a user. Also, a queue will retain a production lock until the user unlocks it or navigates to another queue. For more information, see [“Removing Locks from a Workstation” on page 146](#).

If a queue is mistakenly left with a production lock, it is considered to be in a busy state. You must unbusy the queue from the console before anyone can order it. For more information, see [“Unbusy Stories and Queues” on page 147](#).


Removing Locks from a Workstation

In some cases, you can unlock stories or queues from an iNEWS Workstation. This section provides procedures for removing the following types of locks:

- A story’s edit lock
- A story’s user lock without a key
- A queue’s user lock without a key
- A queue’s production lock

Before you unlock a story or queue, find out who locked the story. You want to make certain that you do not unlock a story or queue that is being used at the time. For more information, see [“Identifying Locked Queues and Stories” on page 141](#).

To remove a story’s edit lock from an iNEWS Workstation:

- ▶ Navigate to another story.
- ▶ Type Ctrl+E, which removes the lock but allows you to stay in the story.
- ▶ Select Story > Edit unlock.
-  ▶ Click the Edit Lock button on the Main toolbar—shown at left. The edit lock is removed when the button no longer appears pushed in.



Of the four possible steps to take, the last three will prompt you to save any changes.

To unlock a story without knowing its key:

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

7. Save the story.

To unlock a queue without knowing its key:

1. At a workstation, log in as a system administrator—that is, with a superuser account. You must be a system administrator to remove a lock for which you do not know the key.
2. Select the queue to unlock.
3. Select Tools > Unlock Queue. A User Unlock Queue dialog box will appear.
4. Click the Unlock button.
5. Click OK.

To remove a queue's production lock from an iNEWS Workstation:

- ▶ Select Tools > Production Lock. A checkmark appearing next to the Production Lock menu option indicates whether the lock is in place.
- ▶ Navigate to another queue. Production Lock is disabled automatically.

Unbusy Stories and Queues

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



Always ensure a lock is invalid first. Determine whether the edit or order lock is not the result of someone actually editing a story or ordering a queue. Do not unbusy a story or queue unless you have determined that the lock is invalid. Unbusing a story or queue that is in use can cause serious problems.

To unbusy a story or queue at the console:

1. Use the unbusy command with the following format: `unbusy <queue name>`

The queue name is the name of the locked queue or the name of the queue containing the locked story.

For instance, if a locked story was in SHOW.SCRIPTS or that queue was order-locked, unbusy it by typing: `unbusy show.scripts`.

2. Press Enter.

3. One of the following will happen:

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

```
'show.scripts' Production locked by news from terminal 607 at
2004-02-28 17:51:46
```

5 The Database: Directories, Queues, and Stories

User news no longer logged in at terminal 607 - Unlock? (n/y/q)

To unbusy the queue, type **y**; otherwise, type **n**.

- The unbusy command checks for an edit-locked story in the queue. If it finds one, it checks the workstation where the story was last edited. If anyone is logged in at that workstation, the console displays the story's form and prompts you to check with the person who is logged in at that workstation.

For instance, if the story, Shootout, was last edited at workstation 4589 and user, Smith, is currently logged in at workstation 4589, the console displays the following:

```
NRCS-A# unbusy show.10pm.rundown
```

```
page-number    A02
presenter      KSA
title          SHOOTOUT
v-cam
v-shot
v-format       W/VO
v-graphics
audio-time     0:30
runs-time      0:00
total-time     0:30
back-time
status         OK
endorse-by
modify-date    2009-03-05 08:38:12
modify-by      smith
v-edby
v-writer
video-id       091907TR10
item-channel   A
event-status
v-tapenumber
v-timecode
air-date
```

Did you check with smith who is at device 4589?

b locked on 2009-05-20 at 16:59:18 by session 4589

Unbusy? (n/y/q)

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



If a story is unbusy at the console while a user is editing it, when the user tries to save the story, the story is saved to the Dead queue, and the user gets an error message that states: Story save failed: Error: Story saved to dead.

5 The Database: Directories, Queues, and Stories

6 Groups

This chapter explains how to create groups in iNEWS and use the system's group-related features to customize system usage. When possible, procedures are explained using the workstation, but some can also be accomplished at the console, with extensive use of the `gtraits` command. This command is similar to the `utraits` command and the `dbtraits` command. All three commands are explained in [“Managing Traits at the Console” on page 623](#).

This chapter also covers viewing and modifying information in the `SYSTEM.GROUPS` queue at the iNEWS Workstation and the `/site/system` file at the console.

This chapter contains the following main sections:

- [Overview of Groups](#)
- [Viewing Group Information from the Console](#)
- [Viewing Group Information from a Workstation](#)
- [Creating a New Group](#)
- [Renaming a Group](#)
- [Deleting a Group](#)
- [Creating or Modifying Multiple Groups](#)
- [Adding Users as Members of a Group](#)
- [Adding Groups as Members of Other Groups](#)
- [Adding Workstations as Members of a Group](#)
- [Combined Permissions and Timeouts](#)
- [Group Access and Usage Restrictions](#)
- [Group Traits for the Database](#)
- [Creating a Mail Alias](#)

Overview of Groups

The iNEWS newsroom computer system lets you categorize users by placing accounts belonging to people with similar needs into the same group. Organizing users in this way enables you to more easily customize your system. It also enables you to apply security restrictions at the group level.

For instance, you can:

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

You can create as many as 700 groups, and you can assign an unlimited number of users to each group.

Viewing Group Information from the Console

At the console, various versions of the `gtraits` command are used for viewing and modifying group information. You must take superuser privileges at the console before using any form of this command, as indicated by the prompt, `NRCS-A#`. See [“Entering Superuser Mode” on page 37](#) for more information.

To get information for a list of all groups in the iNEWS database:

▶ Type: `NRCS-A# gtraits list-`

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

▶ Type: `NRCS-A# gtraits list producers`

To get a list of all groups to which a particular user belongs:

▶ Type the command followed by the user’s ID, such as:

```
NRCS-A# gtraits list jdoe
```

Sometimes, for security reasons, groups are assigned as group traits to directories and queues in the database.

This information can be viewed from the console by using the `list d-v` console command, which lists each queue's assigned read, write, and notification groups.

Use the following format to view a list of group information for a specific queue:

```
list d-g <queue name>
```

Use the following format to view a list of all queues in the database that have a particular group assigned as their read, write, or notification group:

```
list rwng=<group name> d
```



All of the gtraits list variants can be produced using list g.

Viewing Group Information from a Workstation

You can also view group membership and group assignment information from an iNEWS Workstation. However, since groups are created at the console, the information you receive can potentially be out of sync with the information that is actually in the database. Therefore, use this method as a quick way to get information that you recognize may require verification at the console.

To view information about group memberships from a workstation:

1. Locate the SYSTEM.GROUPS queue in the Directory panel.
2. Double-click the Groups queue to open it.

The Queue panel contains a list of the names of the existing groups with the first group name selected. The members of the group appear in the Story panel.

3. To view the contents of different groups, do one of the following:
 - ▶ Use the mouse to click a different group listed in the Queue panel.
 - ▶ Use the Up or Down arrow keys or the scroll bar on the right side of the Queue panel to move to another group.
 - ▶ Use the Page Up or Page Down keys to scroll several groups (up or down) in the list at a time.

To view what group is assigned to a queue or directory for read and write access, or for notification purposes:

1. Locate the queue or directory you want to know about in the Directory panel.
2. Right-click on the queue or directory.
3. Select Properties from the context menu.
4. In the Directory/Queue Properties dialog box, select the Groups tab.

Creating a New Group

Creating a new group is a three-step process. The first part of the procedure must be done at the console. The last is done at the iNEWS Workstation.

To create a new group:

1. Choose a group name.
2. Create the new group at the console.
3. Create a group's membership story in the database and specify members.

Each of these steps are covered in more detail in the next few sections.

Step 1 - Choosing a Group Name

Before you actually create a new group, you need to decide on the name of the group.

To choose a group name:

1. Follow these guidelines:
 - ▶ Group names cannot be more than 20 characters long and cannot contain spaces. A group name longer than 20 characters will be truncated to 20 characters.
 - ▶ You cannot use a name already used as someone's User ID.
 - ▶ Some words are reserved by the system for special purposes, and cannot be used as names for groups. These words—which include: alias, all, group, and restricted—are defined in the Words dictionary by use of the W_ALIAS, W_ALL, W_GROUP, and W_RESTRICTED tokens, respectively.
 - ▶ Choose a name that indicates the purpose or general makeup of the group, for instance, you may want to call the group that includes all your producers by the name “producers.”
2. View the list of groups in the SYSTEM.GROUPS queue to ensure the name you select is not already in use. If a group with your chosen name already exists, determine whether its members are the same as those users you want to assign to your group. The existing group might represent these users, so you can use it instead of creating a new one.

Step 2 - Create New Group at Console

For this procedure, which uses fieldreporters as an example, you must first take superuser privileges at the console before using any form of the gtraits command, as indicated by the prompt, NRCS-A# in the following procedure. See [“Entering Superuser Mode” on page 37](#) for more information.

To enter the new group name in the iNEWS database:

1. Enter superuser mode at the console.
2. Verify that the new group name is not already used by typing the following kind of command at the console:

```
NRCS-A# gtraits list fieldreporters
fieldreporters is not a user or group name
```

In the example, the system response indicates that fieldreporters is not currently a group name. You should receive a similar response before proceeding.

3. Use the following command to create the new group name in the system:

```
NRCS-A# gtraits add fieldreporters
Added group fieldreporters
```

4. Press Ctrl+D to leave superuser mode.

Step 3 - Creating Group's Membership Story and Specifying Members

This procedure, which uses fieldreporters as an example, is done at an iNEWS Workstation. Members of a group are listed in a story that bears the name of the group and is located in SYSTEM.GROUPS. If the story doesn't exist, you must create it as part of this process.

To create a group's membership story and specify members of that new group:

1. Open the System folder from the Directory panel.
2. Open the Groups queue.
3. Do one of the following:
 - ▶ Select File > New Story.
 - ▶ Press the Insert key.

In the Queue panel, a blank row appears in the group list, and a blank story appears in the Story panel.



When your iNEWS newsroom computer system was installed, membership lists for several groups common to newsrooms were placed in the Groups queue. To make it easy to maintain, each list was placed in a separate story. Continue this convention to organize groups. You can put more than one group's membership list (or all of them) in a single story, but it is not recommended.

4. Type the name of the group, such as fieldreporters, in the Title (Slug) field of the Queue panel or in the corresponding field of the Story Form panel.
5. Press Enter.

6 Groups

6. Click inside the Story Text panel and type the group name and membership list in this format:

```
group fieldreporters
user-ID user-ID . . .
user-ID user-ID . . .
```

7. Select File > Save Story.

This procedure creates a story, stored in SYSTEM.GROUPS, that bears the group name and contains the membership list for the group. The system will refer to the story anytime its group is applied to security measures or other system features.

Group Checker

When you save your changes to a group, the system automatically runs the server program known as the group checker, which looks for errors in the stories in SYSTEM.GROUPS. Changes to the SYSTEM.GROUPS queue trigger the group checker because the SYSTEM.GROUPS queue is assigned the Group system mailbox.

The group checker may also be run manually from the console by typing the command, `grpcheck`, using the following format:

```
grpcheck [-v[v]] <name of queue>
```

For instance, to manually start the group checker to check group membership lists in the database, type: `grpcheck -v system.groups`

The group checker may take a minute or two to process. When it finishes, the system sends you one or more messages describing the results.



At an iNEWS Workstation, the system alerts you to the fact that you have received a message by an audio tone and a flashing Message bar button in the menu bar. On the Main toolbar, click the Message bar button—shown at left—to read the messages from the group checker on the Messages toolbar—shown below. The name, `grpcheck`, appears in the From field. The messages appear in the Message field.

If the group checker finds no errors in the story you have created to list the members of the new group, you get a GROUPS story OK message.



If the group checker finds an error, it sends a message indicating the story and the line in that story in which the error occurs. For a complete list of group checker messages, see [“Group Checker Error Messages” on page 157](#).

In some cases, multiple errors are discovered, resulting in several error messages.

To display the entire list of error messages sent to you:



- ▶ Click the History button on the Messages toolbar.
- ▶ Select Communicate > Messages > Show History.

You can also copy or paste the contents of the group checker messages to another file.

If the errors are not serious, the last message is: GROUPS story accepted, with errors. This message indicates that the group checker applied changes that were not in error.

If there are serious errors, the last message is: GROUPS story NOT OK. This message indicates that the group checker could not use any changes because of errors.

The group checker examines whatever work you just completed along with everything else in SYSTEM.GROUPS. This means that if some of the error messages you see are not related to your changes, they are possibly the result of pre-existing errors in a different story in SYSTEM.GROUPS.

Group Checker Error Messages

The following is an alphabetical listing of error messages you may see when you create or edit group membership lists or alias definitions. The group checker usually takes a minute or two to completely process the stories in SYSTEM.GROUPS and report any error messages.

Bad workstation device specification

- You did not enter a workstation’s device number or device name correctly. Usually, this happens because you did not use a closing brace (}) in the declaration.

Cannot open default aliases file

- The group checker could not open an internal file that it uses to check alias entries. Call Avid Customer Support.

Cannot open new aliases file

- The group checker could not create a new aliases file to reflect the changes you made. Call Avid Customer Support.

Cannot save old aliases file

6 Groups

- An internal error occurred. Call Avid Customer Support.

Duplicate group or alias name

- You tried to create two groups or aliases with the same name.

Failed to open queue

- Due to an internal error, the group checker was unable to open SYSTEM.GROUPS. Call Avid Customer Support.

Failed to open story

- Due to an internal error, the group checker was unable to open one of the stories in SYSTEM.GROUPS. Call Avid Customer Support.

Group or alias word missing. Skipping text

- You did not begin a mail alias definition with the word ‘alias’, or a group membership list with the word ‘group’. This is followed by the line number where the group checker expected to find ‘alias’ or ‘group’. The W_GROUP and W_ALIAS dictionary tokens define what words, such as ‘alias’ or ‘group’, are used by the group checker. The tokens may be customized in the /site/dict/words file on the iNEWS Server.



For more information on mail aliases, see “Creating a Mail Alias” on page 176.

GROUPS story accepted, with errors

- Errors appear in the group story, but none are serious. The group checker will use the entries that do not have errors. The entries that have errors are ignored.

GROUPS story NOT OK

- Serious errors appear in the group story, and the group checker cannot use it.

GROUPS story OK

- There are no problems with the group story.

Ignoring words following alias name

- Any words you include on the same line after the name of the alias you are defining are ignored.

Ignoring words following group name

- Any words you include on the same line after the name of the group you are defining are ignored.

Improper use of reserved word

- You cannot use a reserved word, such as 'alias' and 'group', as a group or alias name.

Internal groupchecker error

- Some undefined error occurred while the group checker was running. Call Avid Customer Support.

Invalid name follows word "alias"

- The name of the alias is invalid. It might be too long. Check the alias and remove or correct the user name.

Invalid name follows word "group"

- The name of the group is invalid. It might be too long or not exist on the server. Check that the group has been created on the server and that it matches the name being used in the client.

More than 50,000 alias names created

- The system created many pseudo-alias names to break up individual aliases into lists of 1000 characters or less. Call Avid Customer Support.

Missing alias name

- You did not follow the word alias with the name you want the alias to have.

Missing group name

- You did not follow the word group with the name of a group.

Name already used as alias name

- You created a group with the same name as an alias already defined in the story or queue.

Name already used as group name

- You created an alias with the same name as a group already defined in the story or queue.

No groups or aliases found

- All stories in SYSTEM.GROUPS are empty—they contain no aliases or groups.

Not a workstation device

- You included something in braces ({}) that is not a workstation device name or number. The message is followed by the name you tried to include as a workstation.

Not a user or workstation

6 Groups

- You defined something that is not a recognized user or workstation as a member of a group.

Recursive group membership

- You defined a membership list that created a recursion error.

User name used as group or alias name

- You cannot give a group or an alias the same name as an existing user.

Renaming a Group

Renaming a group is a two-step process. The first step of the procedure must be done at the console in superuser mode. The second step is done at the iNEWS Workstation.

To rename a group:

1. Change the group name in the system's database.
2. Change the group name in SYSTEM.GROUPS.

Each of these steps are covered in more detail in the next couple of sections.

Step 1 - Change Group Name in Database

To change the group name in the system's database:

- ▶ Type the gtraits r command, which has this syntax:

```
gtraits r <old-group-name> <new-group-name>
```

For instance, to change the group name producers to 5pmproducers, type:

```
NRCS-A# gtraits r producers 5pmproducers
```

Renamed producers to 5pmproducers.

Step 2 - Change Group Name in SYSTEM.GROUPS

To change the name in the group's membership story:

1. Locate the group's membership story in SYSTEM.GROUPS.
2. Modify the name of the group, which appears in the title field of the queue and story form, and on the first line of the story.
3. Change the group name if it appears as a member in any other membership stories.



If you do not do step 3, the next time you make a change in the queue, the group checker warns you that the membership list uses an invalid group name. See “Group Checker” on page 156 for more information.

Deleting a Group

Deleting a group involves steps that must be done at the console and the workstation. The console portion of the procedure is a two-stage process. The `gtraits` command you type marks the group for deletion, but the group is actually deleted the next time the system runs the `dbpurge` process, which it does at 15 minutes past every hour.



You cannot use any `gtraits` commands on a group that is marked for deletion but still waiting to be deleted.

To delete a group in the system:

1. Enter superuser mode at the console.
2. Delete the name in the system’s database at the console using the `gtraits d` command, which has this syntax: `gtraits d <group-name>`

For instance, to delete the `5pmproducers` group, type:

```
NRCS-A# gtraits d 5pmproducers
Marked 5pmproducers for deletion.
```

3. Delete the group name and its membership list story in `SYSTEM.GROUPS`.
4. Delete the group name if it appears as a member in any other membership lists.



If you do not do step 3, the next time you make a change in the queue, the group checker spots the deleted group’s membership list and warns you that it uses an invalid group name. If you do not do step 4, the next time you make a change in that membership list story, the group checker warns you that the membership list uses an invalid group name. See “Group Checker” on page 156 for more information.

5. Press `Ctrl+D` to leave superuser mode.

Creating or Modifying Multiple Groups

The `gtraits` console command has an interactive mode in which you can execute group-related commands without entering `gtraits` each time. Use this mode when there are a number of `gtraits` commands you want to enter in succession.

To enter the gtraits interactive mode:

1. Enter superuser mode at the console.
2. Type:

```
NRCS-A# gtraits i  
>
```

The command replaces the normal system prompt with an angle bracket (>) to indicate that you are in interactive mode. At this prompt, you can enter any gtraits command, such as `change group` or `add`, without typing the gtraits command for each operation.

For instance, to add the `5pmproducers` group, type:

```
> add 5pmproducers
```

If you were not in interactive mode, the entire command line would be required, such as: `gtraits add 5pmproducers`.

3. Type `quit` or `q` to leave interactive mode.
4. Press `Ctrl+D` to leave superuser mode.

Adding Users as Members of a Group

There are three possibilities for membership in a group:

- An individual user can be a member of a group.
- One group can be a member of another group (making all the users of the first group members of the second).
- A workstation can be a member of a group.

This section provides the procedure for adding a user to a group. You must be at an iNEWS Workstation to change group membership.

To add individual users to an existing group:

1. Open the System folder from the Directory panel.
2. Open the Groups queue.
3. Locate the story or stories in your `SYSTEM.GROUPS` queue containing the group membership list you want to modify or add members to.
4. Open that story in the Story panel.

5. Type in the user ID(s) you want to add to the group. The user IDs need not be on the same line as the group name. For instance, add user sjones to the fieldreporters group:

```
group fieldreporters
jdoe bsmith ftaylor
ghendrickson jgrey sjones
```

6. Select File > Save Story.
7. Verify the approval of the story from the messages sent to you from the system's group checker server program.

Adding Groups as Members of Other Groups

In addition to adding individual users to groups, you can add an entire group to another group. The members of the first group become members of the second group. The order in which you define groups in your SYSTEM.GROUPS queue is not important; however, care should be taken to avoid recursion. This section provides the procedures for adding a group to another group and avoiding recursion.

You must be at an iNEWS Workstation to change group membership.

To add a group to an existing group:

1. Open the System folder from the Directory panel.
2. Open the Groups queue.
3. Locate the story or stories in your SYSTEM.GROUPS queue containing the group membership list you want to add or modify.
4. Open that story in the Story panel.
5. Type in the group name you want to add to the group.
6. Select File > Save Story.
7. Verify the approval of the story from the messages sent to you from the system's group checker server program.

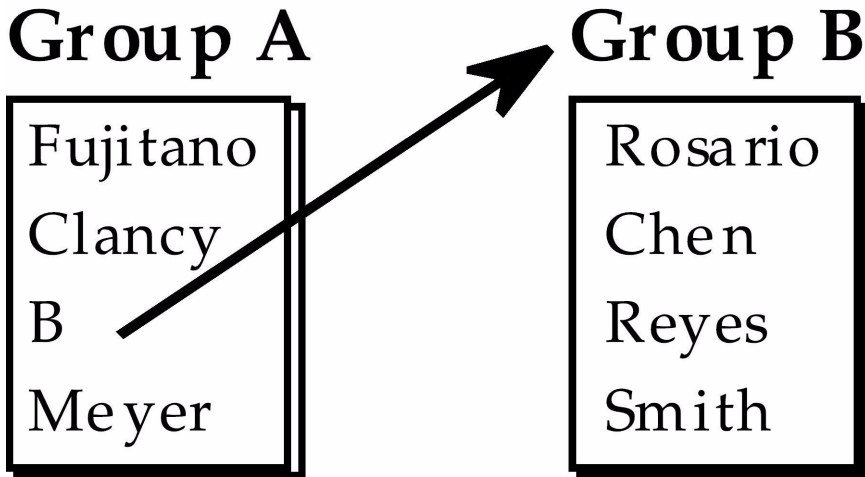
Avoiding Recursion

When you make groups members of other groups, do not create a membership list that contains a circular reference, also called recursion. The following is an explanation of recursion in a membership list.

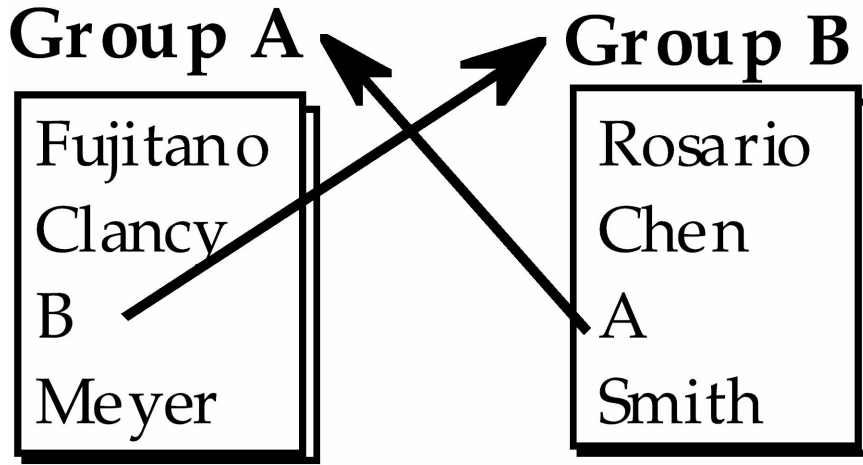
6 Groups

When the group checker examines a group's membership list story, it builds an internal list of the group's members. If one member is the name of another group, the group checker must determine the members of the second group and add them to the first group's internal membership list.

For instance, in the following diagram, Group B is a member of Group A. When the group checker evaluates the membership list for Group A, it creates an internal membership list for Group A that contains users Fujitano, Clancy, Meyer, Rosario, Chen, Reyes, and Smith. This example is not recursive and causes no problems for the group checker.



Recursion occurs when the group checker cannot resolve memberships because one group in the chain refers to another group higher up in the chain. For instance, a case of recursion would occur if Group B is a member of Group A, but Group A is also a member of Group B.



The group checker cannot create internal membership lists for these groups. When it evaluates Group A, it sees that Group B belongs to Group A and tries to add B group's members to A group's internal list. However, one of the Group B members is Group A, which the group checker has still not resolved. The group checker cannot proceed.

If you see a recursion error message, examine your membership lists for incidents like this and remove the recursive reference. In the previous example, either remove Group B from the membership list for Group A or Group A from the membership list for Group B.

To check for recursion at the console:

► Type: NRCS-A# `grpcheck -v system.groups`

The group checker displays the title of the story in which it finds an error and a description of the error. For instance, after typing the `grpcheck -v` command, the following output shows a story with the title `group_249` that has recursive entries in its membership list:

```
grpcheck: 09:09:13 [CONSOLE] [group_249] Recursive group membership
a->b->a
grpcheck: 09:09:13 [CONSOLE] [group_249] GROUPS story NOT OK
```

Adding Workstations as Members of a Group

Suppose you have a workstation that is used by several staff members, all of whom are producers. When they use this workstation, they need access to the queues that members of the producers group normally have access to. When they are using other workstations, they do not need the special producer privileges.

You can grant the producer workstation the security permissions granted to the producer group by adding it as a member to the group called producers in SYSTEM.GROUPS.

You must be at an iNEWS Workstation to change group membership.

To add a workstation to an existing group:

1. Open the System folder from the Directory panel.
2. Open the Groups queue.
3. Locate the story or stories in your SYSTEM.GROUPS queue containing the group membership list you want to modify or add members to.
4. Open that story in the Story panel.
5. Type in the workstation's session number in braces to add it to the group.
6. Select File > Save Story.
7. Verify the approval of the story from the messages sent to you from the system's group checker server program.

When a user logs in at a workstation, the system ordinarily combines any system permissions the user already has with permissions the workstation may have. For instance, a user belonging to the writers group who logs in at a workstation assigned to the producers group would have access to directories and queues accessible to both groups.

System permissions that apply to workstations are assigned using the security parameter that is set in the /site/system file. The security parameter in this file is either OR or AND. OR security uses the security level set for either the user or the workstation. AND security uses the security level set for both the user and the workstation. For more on these security parameters, see [“Combined Permissions and Timeouts” on page 167](#). For more information on how to edit the /site/system file, see [“System Configuration” on page 243](#).

A workstation can only be used as a member of a group if that PC is assigned a dedicated session in the iNEWS configuration file. For instance, an iNEWS Workstation, identified as 319, is assigned a certain group membership. Its configuration line appears with a specified session number, but is not dedicated to any particular PC, as shown:

```
inws 319 - gnews - ;nondedicated
```

The session, 319, is nondedicated, meaning that any PC in the newsroom may be logged in on session 319, and therefore view whatever area is restricted to the group 319 is assigned to. To restrict a queue or directory to a specific workstation, such as 319, then in addition to adding that workstation's session number to the group's membership, the line in the configuration file must be dedicated to a specific PC's IP address.

```
inws      319      192.168.20.136      gnews    -      ;nondedicated
```



Using session numbers in groups will provide proper security only if you have dedicated resources locking down specific PCs to specific resources in the configuration file. This requires that PCs have set IP addresses and would not work in a DHCP environment, where a PC may not always receive the same IP address.

MAC addresses cannot be used directly. The IP or MAC is assigned to a session in /site/config, such as:

```
inws      4595      00:19:B9:0E:7D:E1      gnews    -      ; IT-laptop-01
inws      4596      172.24.96.247      gnews    -      ; IT-desk-01
```

Then the session number would go into the group, such as:

```
group ipmac
{4595}
{4596}
```

See [“Group Access and Usage Restrictions” on page 169](#) for more information.

Combined Permissions and Timeouts

Combining group and workstation permissions enables you to choose to apply additional security to your system, or less security:

- For additional security, you can specify that users at a particular workstation have membership in both human and workstation groups to perform certain actions.
- For less security, you can force the system to check only whether a user at a particular workstation is a member of either the human or the workstation group.

More Restrictive

You can be more restrictive about the permissions granted to users if you include the security parameter in your system profile—that is, the /site/system file—and assign it the value of AND.

6 Groups

For instance, you can use this security level for read permissions on a queue that should be read only by producers that are both members of the producers group and are sitting at the producers workstation.

Less Restrictive

Omitting the security parameter from the system profile or assigning it a value of OR indicates that a user on a workstation is considered to be in a certain group if either the user or the workstation is a member of the target group.

Automatic Timeout

Either type of security described previously works well only if users log out from their workstations when they leave their desks. Otherwise, anyone with access to the workstation where a user logged in can take advantage of that user's or workstation's permissions.

You can prevent this by setting your workstations to automatically log out after a certain period of inactivity. To do that, you need to edit one or more special timeout parameters in the system profile at the console. See [“System Configuration” on page 243](#) for more information on how to edit the system profile.

There are two types of timeout parameters:

- Idle timeout
- Login timeout

Idle Timeout

The most useful kind of automatic timeout is the idle timeout, which logs out a workstation if no activity has taken place on it in a specified length of time.

Modify the `localtimeout` value in `/site/system` to change this value for local workstations (those that are most likely within the newsroom).

Modify the `remotetimeout` value in `/site/system` to set the idle timeout value for dial-up workstations.

Timeout Value Settings

All four timeout parameters accept values in the `mmm:ss` (minutes and seconds) format. The following table lists the parameters, the types of workstations they affect, and their maximum and default values.

| Parameter | Workstation Types | Maximum | Default |
|---------------|---|---------|---------|
| localtimeout | Serial, network, PCs | 540:00 | 00:00* |
| remotetimeout | Dial-up (line devices and connect sessions) | 540:00 | 00:00* |

* *Disables timeouts*

Group Access and Usage Restrictions

The iNEWS newsroom computer system is designed to be used by a large group of people, ranging from temporary writer interns to technical producers. To ensure the correct level of security on a system that is accommodating such a wide range of capabilities and responsibilities, restrict access to sensitive areas of the database to people with a need to access the information.

The iNEWS newsroom computer system has security features that let you provide these kind of access restrictions. For instance, you can assign groups to a queue as a read and/or write group trait. By doing so, you can control which users can read and/or write stories in that queue.

If you do not assign groups to a directory or queue as read and write group traits, the directory or queue is available to all users who are not in the “restricted” group.



System administrators can create a group called “restricted” that can be used to limit its members accessibility to areas of the database.

Access and Usage Examples

Here are some other examples of how access is modified based on group trait assignments:

- If both read and write groups are assigned to a queue, and a user’s ID does not appear as a member of either group, the that user will be unable to create a story in that queue. In other words, the user will not have read and write permission to that queue.
- A user also needs both read and write permission to lock or unlock a queue.
- A user who has only write permission to a queue can copy or move a story into that queue; however, without read permission, the user will be unable to see the queue and its contents in the Directory panel.

6 Groups

- It is possible to change a directory’s read or write group, and thereby modify the read-and-write permissions for all the stories in all the queues in that directory.
- Stories that you move or duplicate into a queue whose general trait is turned on retain their original security. For instance, the Dead queue usually has this trait turned on so stories moved there retain the read-and-write restrictions from their original queues.

Group Traits for the Database

There are five group traits that can be assigned to queues and directories in the database:

- General
- Read Group
- Write Group
- Notify Group
- Editorial Group



You must be a system administrator—that is, logged in with a superuser account—or know the database manager password to modify any trait in the Directory/Queue Properties dialog box. See “Directory/Queue Properties Dialog Box” on page 122 for more information.

All of the group traits can be assigned at the console and the iNEWS Workstation. For procedures at the console, see “Managing Traits at the Console” on page 623 for more information. Procedures at the workstation, using the Groups tab in the Directory/Queue Properties dialog box, are covered in this chapter.

The screenshot shows a dialog box titled "Queue Properties" with a close button (X) in the top right corner. The queue name "[NRC\$]SHOW.10PM.RUNDOWN" is displayed in a text field at the top. Below the title bar, there are three tabs: "User Interface", "Interplay Server", and "Locks". The "Interplay Server" tab is selected, and within it, the "Groups" sub-tab is active. The "Forms" and "Maintain" sub-tabs are also visible. Under the "Groups" sub-tab, there is a checkbox for "General" which is currently unchecked. Below this, there are four text input fields for group assignments: "Read Group" (containing "!<none>"), "Write Group" (containing "reporters"), "Editorial Group" (containing "!<none>"), and "Notify Group" (containing "producers" and highlighted with a blue selection box). At the bottom of the dialog box, there are "OK" and "Cancel" buttons.

The General trait, when applied to a queue, means that stories moved to the queue will retain their original security restrictions, as set by the read and write group traits. This will prevent any unintentional accessibility to stories moved from a highly secure queue to one widely accessible to users.

The other four group traits (Read, Write, Notify, and Editorial) restrict who can read or write stories in a queue; who can remove, break, and float stories in a queue; and indicate who is notified when stories are changed in it. Each of these group traits is explained in the following sections. See “Groups Tab” on page 128 for more information.



The General trait would normally be assigned to the Dead queue.

When a group assignment on a queue is changed from the graphical user interface (GUI), the group security assigned to the stories in the queue is changed and the system automatically re-applies the security change to each individual story. If there are many stories in the queue, the client will remain unresponsive while the new security setting is adjusted for each story and control will not be returned to the user until it is done.



If the queue has the General trait, changing the group security on the queue does NOT change the security assignment on the pre-existing stories created under the old group assignment.

Read Group

A directory or queue’s read group specifies who can read stories in the queue. Users who are not in the read group for the directory or queue cannot see the directory or queue in the file structure displayed in the Directory panel.

To assign a group as a read group to a queue or directory:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the context menu.
3. Select the Groups tab.
4. Select a group from the Read Group drop-down list. Only groups that are already created in the system database will appear in the list. When !<none> is selected, no group is applied; therefore, all users will have read access to the queue or directory.
5. Click OK to save settings.

To remove a group as a read group from a queue or directory:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the context menu.

6 Groups

3. Select the Groups tab.
4. Select !<none> from the Read Group drop-down list. When !<none> is selected, no group is applied; therefore, all users will have read access to the queue or directory.
5. Click OK to save settings.

Write Group

A queue's write group specifies who can add or modify stories in the queue.

To assign a group as a write group to a queue or directory:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the context menu.
3. Select the Groups tab.
4. Select a group from the Write Group drop-down list. Only groups that are already created in the system database will appear in the list. When !<none> is selected, no group is applied; therefore, all users will have write access to the queue or directory.
5. Click OK to save settings.

To remove a group as a write group from a queue or directory:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the context menu.
3. Select the Groups tab.
4. Select !<none> from the Write Group drop-down list. When !<none> is selected, no group is applied; therefore, all users will have write access to the queue or directory.
5. Click OK to save settings.

Notification Group

A queue's notification group specifies which users are notified whenever stories are added or modified in the queue.

To assign a group as a notification group to a queue or directory:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the context menu.
3. Select the Groups tab.

4. Select a group from the Notify Group drop-down list. Only groups that are already created in the system database will appear in the list.
5. Click OK to save settings.

To remove a group as a notification group from a queue or directory:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the context menu.
3. Select the Groups tab.
4. Select !<none> from the Notify Group drop-down list. When !<none> is selected, no group is applied; therefore, no users will be notified whenever modifications are made to the queue or directory.
5. Click OK to save settings.

Editorial Group

A queue's editorial group specifies who can add breaks, float stories, or reorder queues, as well as delete, remove or move data from a queue or directory. All editorial actions require write access.



When an editorial group is specified, users with Write privileges can still create, add, copy and move stories to the queue, and they can edit stories in the queue.

To assign a group as an editorial group to a queue or directory:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the context menu.
3. Select the Groups tab.
4. Select a group from the Editorial Group drop-down list. Only groups that are already created in the system database will appear in the list. When !<none> is selected, no group is applied; therefore, only a user who has Write privileges to the queue can add breaks or float, reorder, delete, or move stories from the queue.
5. Click OK to save settings.

To remove a group as an editorial group from a queue or directory:

1. Locate the directory or queue you want to modify in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder or queue in the Directory panel and selecting Properties from the context menu.
3. Select the Groups tab.

6 Groups

4. Select !<none> from the Editorial Group drop-down list. When !<none> is selected, no group is applied; therefore, only a user who has Write privileges to the queue can add breaks or float, reorder, delete, or move stories from the queue.
5. Click OK to save settings.

Restricted Group

Members of the restricted group are prevented from accessing directories and queues that have no group traits assigned. For instance, if you set a queue's Read and Write group settings—located on the Groups tab of the Directory/Queue Properties dialog box—to !<none>, then that queue has no Read or Write group assigned; therefore, a member of the restricted group will not be able to see or write to that queue. The restricted group is commonly used at sites for temporary users. The system administrator can assign a temporary user to the restricted group and then to a limited set of specific groups. That user is then restricted to only those directories and queues that have those specific groups.

Restricting Both Reading and Writing

You may need to restrict a queue so that one group of users can read and write in that queue, while another group can only read stories.

Suppose you want to restrict your Assignments directory. In most systems, a few people—mostly those at the assignments desk—need write permission to this directory. A larger number of users, such as writers and reporters, need to read, but not edit, stories in the Assignments directory.

The people who should have read-and-write permission for the Assignments directory come from different areas of the newsroom, so it is unlikely a group exists with just those users. However, you could set it up like this:

- Create a group called assignments to represent users who need write permission for the Assignments directory.
- Similarly, create a group called staff to represent users who need read permission.
- Assign the staff group to the directory's read group trait and the assignments group to the directory's write group trait.

Transferring Group Assignments

You might need to locate every instance where a particular group is assigned to a directory or queue and change that assignment so that another group is assigned to that directory or queue.

To transfer group assignments:

- ▶ Use this form of the gtraits console command:

```
gtraits transfer <current-group-name> <new-group-name>
```



Groups are marked for transfer, but no changes are made to any directories or queues until dbpurge runs. Both groups that you include in the gtraits transfer command must already exist.

Hiding Queues and Directories

In addition to restricting access to various queues, you can use group access and usage restrictions to hide queues or directories by placing a strict read restriction on them.

A number of queues on your system probably have very tight write security to ensure that only certain users can create and edit stories in those queues. If other users do not need to read the stories in the queue, you may give the queue tight read security. This prevents the queue from appearing on unauthorized users' screens. Some examples of this are the Dead queue, a Suggestions queue, an Employee Evaluations queue, and so forth.



All users that you want to have the capability to send stories to these queues need to have write access to the queue, but not necessarily read access.

Another example is the System directory, which is usually restricted so that only superusers can write stories there. You can hide this directory so that it does not appear in the main directory for normal users by setting its read group to a group that has no users; this group is typically called Nobody. Because superusers can read everything in the database, they can still see the directory.



Many sites put the system administrator's name in the sysop group for e-mail purposes.

For this example, your system could use an empty group called sysop, which is then assigned as the read group trait for the System directory.

To set the System directory's read group to sysop:

1. Locate the System directory in the Directory panel.
2. Open the Directory/Queue Properties dialog box by right-clicking on the folder in the Directory panel and selecting Properties from the context menu.
3. Select the Groups tab.
4. Select a sysop group from the Read Group drop-down list. If it does not appear in the list, it has not been created in the system database. See [“Creating a New Group” on page 154](#) for more information.
5. Click OK to save settings.

Creating a Mail Alias

A mail alias is a name up to 20 characters long that represents a group of people who often receive similar mail. Each mail alias acts like a distribution list. This way, instead of sending mail to each user individually, you can send mail to the alias and the mail server distributes a copy of the mail story to each user on the group's membership list.

Like groups, mail aliases are defined in stories in the Groups queue in the System directory.



You can have an unlimited number of aliases, but only 700 groups. Groups should primarily be used for security purposes; Aliases are not used for security purposes, only for mail delivery.

To create a mail alias, do the following at an iNEWS Workstation:

1. Open the System directory in the Directory panel.
2. Open the Groups queue.
3. Select File > New Story.

A new blank Story panel appears. In the Queue panel, a blank entry in outline appears.

4. Type the name of the alias in the blank field of the Queue panel or in the corresponding title field of the Story Form panel.
5. Click inside the Story panel and type the alias name and membership list in this format:

```
alias alias-name
user-ID user-ID group-name ...
user-ID alias-name ...
```

An alias's membership list must begin with the word `alias` followed by the name of the alias and one or more lines that list user IDs, groups, or aliases that you want to include.

6. Select File > Save Story.



While using aliases for mail distribution, the removal of a user's account may result in mail bounce back errors if that user's name remains listed in `SYSTEM.GROUPS` stories. Remember to remove users names from stories in `SYSTEM.GROUPS` before removing the user accounts from the system. This will reduce the potential for mail bound back errors.

Mail Aliases for Other Machines or the Internet

You can send mail to a user on another system connected to your system over the network. When you put a network mail address in a mail story's TO field, the mail server routes the mail to the correct address, which depending on your setup can be another machine on your iNEWS system or anywhere on the Internet.

You can make it easier to use network mail addresses by assigning them mail aliases. Then, when you want to mail to someone who is not on a system connected to yours, use the mail alias.



Mail in iNEWS cannot be used to send to external e-mail addresses that include underscores; the sending user receives a “Returned Mail” message.

To assign a mail alias to a network mail address, do the following at an iNEWS Workstation:

1. Open the System directory in the Directory panel.
2. Open the Groups queue.
3. Select File > New Story.

A new blank Story panel appears. In the Queue panel, a blank entry in outline appears.

4. Type the name of the alias in the blank field of the Queue panel or in the corresponding title field of the Story Form panel.
5. Click inside the Story panel and type the alias name and membership list in this format:

```
alias alias-name
network-address ...
```

For instance, to assign the network address jan@kbba.com to a mail alias called Jan, type:

```
alias Jan
jan@kbba.com
```

6. Select File > Save Story.

7 Keyboards and Macros

Macros are time-saving routines you can assign to programmable key(s) on your computer keyboard, which then can be invoked with one or two simple keystrokes, or with the click of a button on a customized toolbar. A single macro can be the shortcut to an entire command sequence. They are commonly used for scripting, printing, navigation, floating stories, and so forth.

This chapter contains the following main sections:

- [Types of Macros](#)
- [Creating a Keyboard](#)
- [Creating Macros](#)
- [Keyboard Checker](#)
- [Assigning a Default Keyboard to a User Account](#)

Types of Macros

There are two types of macros.

- A macro assigned to a key or keystroke combination is known as a *keyboard macro*.
- A macro assigned to a customized toolbar button is known as a *toolbar macro*.

Each keyboard macro lets a user enter many characters and commands with the press of a single key or keystroke combination on an actual keyboard. In most cases, the system administrator will create multiple keyboard macros as a set of macros called a *keyboard* that is stored in the system directory. Each keyboard typically contains keyboard macros grouped according to a specific job or task, such as a producer or writer.



Once created, a keyboard can then be selected as part of the user's preferences.

For instance, suppose a writer in the newsroom frequently writes scripts for a particular reporter. The system administrator can assign the text in that reporter's usual sign-off to a macro key included in a set of macros called the Writer keyboard. Then the writer, with that keyboard chosen as a user preference, could put the entire text of the reporter's sign-off in stories with a single keystroke. This same key may be associated with a different macro, such as one that opens a rundown queue, if the user is a producer using a Producer keyboard—a different set of macros.

The system administrator can assign macros to function keys, such as F1, F2, F3, and so on, across the top of the keyboard and to numeric keys on the numeric keypad, located at the right side of the keyboard.



Some function keys are predefined to standard Windows-based operations, such as F5 to refresh the screen or F1 to open an application's help system. While redefining these predefined keys is allowed, it is not recommended. If you save a macro for a key that has a predefined function, the system displays a warning message stating that a reserved key has been redefined. See “Warning Messages” on page 192 for more information.

Other keys known as state keys, such as Control (Ctrl) or Alt, can also be used in conjunction with the function and numeric keypad keys as shortcuts to entire command sequences. Shift is a state key commonly used with function keys, but not with the numeric keypad.

Users can also create their own toolbar macros—using techniques similar to those used by system administrators who write keyboard macros—and assign them to customized toolbar buttons.

Creating a Keyboard

An iNEWS Workstation keyboard can contain more than 100 macros, representing possible states of the 12 function keys plus the 10 numeric keypad keys. For example, possible combinations for F7 are:

| | |
|----------|-------------------|
| F7 | Ctrl+Shift+F7 |
| Shift+F7 | Alt+Shift+F7 |
| Ctrl+F7 | Alt+Ctrl+F7 |
| Alt+F7 | Alt+Ctrl+Shift+F7 |

Keyboards, or sets of keyboard macros, are actually stories saved in a specific location in the iNEWS system directory.

To create a keyboard containing a set of macros for iNEWS Workstations:

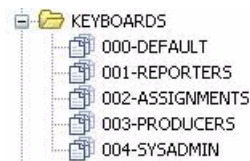
1. Log in to an iNEWS Workstation.
2. Navigate to SYSTEM.KEYBOARDS in the Directory panel.
3. Create a new queue to hold the story that will contain your keyboard macros.

Be sure to name the queue according to the following format:

<###>-<keyboard name>

The numerical values can be any three-digit number from 000 to 255. If the number is less than 100, you must supply leading zeroes. The keyboard name is the name that will appear as a keyboard available for selection as a user preference.

Some sample names for queues in SYSTEM.KEYBOARDS are: 001-Reporters, 002-Assignments, and 003-Producers.





Although the numerical values of 251-255 are valid, you cannot have more than 250 queues in `SYSTEM.KEYBOARDS`, since this would exceed the limit on the number of queues allowed in a single directory. Queues with the number 256 or greater will not work.

4. Open the new queue.
5. Press the Insert key to create a new story in the queue.

If an existing keyboard story is similar to the new one you want to create, you can copy it into the queue and modify it rather than creating a keyboard story from scratch.

The first story in the queue with a name containing the string `inws`, which is not case-sensitive, is used as the keyboard macro definition story for iNEWS. For instance, a default keyboard story name is: `000 inws DEFAULT`.

| SLUG | CREATED | MODIFIED |
|------------------|-------------------|--------------------|
| 000 inws DEFAULT | 1/9/1998 21:26:45 | 6/22/2006 15:53:08 |

Other examples are: `inws 001 REPORTERS` or `inws 002 ASSIGNMENTS`. As the examples show, the `inws` string can begin the story's name or be somewhere in the name.



Without the `inws` string in a story's name, the keyboard will not work.

6. Create your keyboard macros.

The keyboard macros are listed as text in the story. Keyboard macros for iNEWS are written using the key names which are in the file: `/site/dict/keymacros`. You can use alphabetic and numeric keys, and most punctuation marks in a macro. However, some punctuation marks are reserved for specific functions within a macro. See [“Characters and Keys Available for Macros” on page 185](#) for more information.

7. Save the story.



The “key” mailbox is used by the iNEWS keycheck program, which checks for errors immediately after a keyboard story is saved. See [“Keyboard Checker” on page 189](#) for more information. The “key” mailbox must be set on keyboard queues. The mailbox trait can be viewed and changed from the Directory Properties dialog box. See [“Directory/Queue Properties Dialog Box” on page 122](#) and [“Maintain Tab” on page 130](#) for more information.

Creating Macros

There are numerous things to keep in mind when creating macros. These include the function of the macro (printing, scripting, navigation, etc.), the keystrokes to produce the macro, and the proper macro syntax for creating it.

Keyboard Macros

Keyboard macros begin with the "at" symbol (@) and are written in segments, which make up a *macro definition*. Segments include a Key Indicator, a Separator symbol known as a tilde (~), an Action, and an optional Comment. Segments must appear in the proper order for the macro to work correctly.

For instance: `<Key Indicator> <Separator> <Action> <Comment>`

An example of a macro definition is:

```
@{f4}~{alt gd} wires.all{enter};Go to wires.all
```

This macro allows you to press F4 to navigate to the WIRES.ALL queue rather than completing the longer process—typing Alt+G+D to open the Destination dialog box, then typing **WIRES.ALL**, and pressing Enter—to do the same thing.

Each segment of the sample macro definition are explained in this table:

| Macro Segment | Function |
|--------------------------|---|
| @{f4} | The key indicator begins the macro definition line with the at symbol (@). Then, {f4} indicates the macro is invoked when a user presses and releases the key(s) defined within the braces—in this case, the F4 key. Braces ({ }) are used to group letters together as one key or combination of keys. |
| ~ | The separator is the tilde character (~), and it divides the key indicator you are defining from the action the macro is to perform. |
| {alt gd}wires.all{enter} | The action includes a series of keystroke combinations. In the example, {alt gd} presses the Alt key, then types gd for the Go To, Destination menu options, and releases the Alt key. That part of the action opens the Destination dialog box. Then, wires.all is typed in the text field of the dialog box, and {enter} presses the Enter key and releases it. |
| ;Go to wires.all | The optional comment begins with the semicolon (;) and provides a description of what the macro does. |

7 Keyboards and Macros

Only certain keys can be defined as indicators for macros. In most cases, you can assign macros by combining a function key or numeric key on the keypad with one or more state keys. See [“Using the State Keys in Macros” on page 187](#) for more information.



To use the numeric keys on the keypad, Num Lock must be on.

Other keys can appear in macros but not as indicators—that is, they cannot have macros assigned to them. See [“Characters and Keys Available for Macros” on page 185](#) for more information.

Toolbar Macros

Toolbar macros use only the Action section of the macro definition. The Action is entered in the Customize Toolbar Button dialog box, in the Macro section. For instance:

```
{alt gd} wires.all{enter}
```

Here are some other examples of toolbar macros:

- `{alt sf}` — This is a macro that creates a “float” button for a producer, used to float a story in the rundown.
- `{alt tu}<name of plugin>{enter}` — This is a macro that opens a plugin in iNEWS.
- `{alt osu}` — This is a macro that can be used when copying wires into a story. It puts the text in upper case.
- `{tab}{home}{tab}{tab}{tab}ASSIGNID{enter}{up}` — This is an example of a macro that could be used to enter ASSIGNID into a certain field in a rundown. The macro first exits the form you might be in, tabs over to position the cursor in the field—this will vary depending on the rundown—and types ASSIGNID. The `{up}` at the end is to take your cursor back to the field.
- `{alt fpp}{alt q}{alt y}rundown{enter}` — This macro prints a rundown.
- `{alt{right}}{alt{right}}{alt{left}}{alt vuf}editor{enter}` — This macro, first, ensures the focus is in the Queue panel and then changes views.



Be careful when creating macros, like Assign ID or those used for scripting, that the focus is in the proper panel; otherwise, the macro could damage stories or rundowns.

To name a few more of the many possibilities, toolbar macros can be used to create buttons that will duplicate a story to a queue, insert common production cues in a story, or write a validation code in a field and send the story to a validation server.

Characters and Keys Available for Macros

You can use alphabetic and numeric keys, and most punctuation marks in a macro. However, some punctuation marks are reserved for specific functions within a macro. For instance, because the open and close braces ({ }) and the tilde (~) are characters that have special meanings, you cannot use them as plain text in a macro. You can use the at symbol (@) in the Action of the macro—to the right of the Separator—but not as plain text on the left-side of the Separator, where it indicates the start of a new macro definition.



Do not use the semicolon character in a macro. If you include a semicolon (;) in a description line, everything following that character is considered a comment by the system and is ignored. An example of semicolon use for comments is shown in “Using the State Keys in Macros” on page 187.

If you create a macro longer than 80 characters, let the system wrap the cursor around to the next line.

The following table offers a quick reference to what keys can be used in macros. The function, keypad, and state keys can be used as indicators and, therefore, have macros assigned to them. Edit, arrow and miscellaneous keys can appear in the action part of a macro, but cannot have macros assigned to them.

| Function Keys | Keypad Keys | State Keys | Edit Keys | Arrow Keys | Misc. Keys |
|---------------|-------------|------------|-----------|------------|------------|
| f1 | kp0 | shift | insert | up | tab |
| f2 | kp1 | ctrl | home | down | backspace |
| f3 | kp2 | alt | pageup | left | space |
| f4 | kp3 | | pagedown | right | enter |
| f5 | kp4 | | end | | |
| f6 | kp5 | | | | |
| f7 | kp6 | | | | |
| f8 | kp7 | | | | |
| f9 | kp8 | | | | |
| f10 | kp9 | | | | |
| f11 | | | | | |
| f12 | | | | | |

7 Keyboards and Macros

For more information about how to use state keys, see [“Using the State Keys in Macros”](#) on page 187.

Predefined Function Keys

Some function keys have predefined system functions, such as F1, which opens the iNEWS help system. These keys are provided as accelerator keys for common user functions.



Redefining these predefined keys is allowed, but is not recommended. If you save a macro for a key that has a predefined function, the system displays a warning message stating that a reserved key has been redefined. See [“Warning Messages”](#) on page 192 for more information.

The following table shows the standard predefined system function keys.

| Function Key or Key Combination | Predefined System Function |
|--|---|
| F1 | Opens the iNEWS help system |
| F2 | Edit a field (or cell) in the Queue panel |
| F3 | Find Next |
| Alt+F4 | Exits the iNEWS program |
| Ctrl+F4 | Closes a workspace |
| F5 | Refreshes display in Queue or Story panel |
| Ctrl+F5 | Discard changes |
| F6 | Toggles between the Instruction panel (containing production cues) and the Story Text panel |
| Shift+F6 | Toggles between the Story Text and Story Form panels. |
| F7 | Opens Wire Urgent Workspace (Wire Priority queue) |
| Shift+F7 | Opens Wire Alert History window |
| F8 | Toggles Message toolbar on and off |
| Shift+F8 | Opens Message History dialog box |
| F9 | Toggles Mail Workspace open and closed |

Using the State Keys in Macros

The Alt, Shift, and Ctrl keys are known as *state keys*, because their state affects what happens when another key is pressed, whether they are pressed or not.

To include a state key in a macro:

1. Begin with an open brace ({) to indicate that a key is being pressed.
2. Follow the open brace with the name of the state key, such as Alt.
3. Since a state key does not do anything by itself, enter the name of the next key pressed along with it. Enclose this key in another set of braces ({ }) if it is a function or numeric keypad key, such as F7 or kp1.
4. End with a closing brace (}) to indicate the release of the state key. So you'd have something similar to this: {alt{f7}}.



The iNEWS newsroom computer system will not recognize the Shift state key used in combination with the numeric keypad keys. For instance, {Shift{Ctrl{kp9}}} is the same as {Ctrl{kp9}}. If a user attempts to create a macro with this combination of keys, the system's keycheck program will issue an error message: (M_NUMKEYNOSHIFT) "Shifted numeric keypad 0-9 keys cannot be assigned."

Here is an example of a macro that uses state keys for scripting. Its function is to insert a TAKE VO production cue into a story when the user presses Shift+F1. The first line is a comment. The second line, which wraps in this guide but might not do so on your screen, is the macro.

```
;SH F1 TAKE VO
@{shift{f1}}~{alt{insert}}TAKE VO{f6}{ctrl{alt p}}TAKE VO{ctrl{alt n}}{enter}
```

Using Plain Text in Macros

Besides using individual keys and key combinations in a macro definition, you can also have the macro enter plain text. This could be text you include in stories often or text you enter in fields of a dialog box in iNEWS. Example:

```
@{ctrl{f9}}~{space} Roll tape - Sound up full {space}{enter}
```

Whenever you include plain text in a macro, all spaces in the text are preserved. The case (lowercase or uppercase) is preserved as well.

Repeating Macros

If there are actions performed at the workstation that require executing the same command or series of commands over and over, you can create a repeating macro that performs this action. Once invoked, a repeating macro executes at regular intervals until the user presses Ctrl+Break or Escape.

To have a macro in an iNEWS keyboard perform a repeating function, place the command {repeat} just before the actions you want the macro to repeat.

As an example, create a macro that makes it easier to browse wires and assign it to the F4 key. This macro takes the user to WIRES.ALL and then scrolls down one story at a time. It pauses briefly on each story, so the user can read the title and decide whether to read the story. The user stops scrolling by pressing Ctrl+Break or Escape.

To create this repeating macro:

1. Begin the macro by typing `@{f4}~` to indicate that it is for the F4 key.
2. After the Separator (~), type `{alt gd} wires.all {enter}` to open a window displaying WIRES.ALL.
3. Add the repeating portion of the macro, which moves the user down one line at a time at regular intervals. Begin by typing `{repeat}` to indicate that what follows is repeated. Then type `{down}` to move the cursor down one line in the queue.
4. Use the `{pause}` command to make the macro pause a few seconds on each story. Follow this command with the number of seconds you want the repeating macro to pause. For instance, to make the macro pause two seconds before repeating, type `{pause 2}`. The range for the `{pause}` command is 1 to 60 seconds.

The macro should be one continuous line of text. Otherwise, allow the computer to wrap the text if it extends beyond screen margins.

The completed macro looks like this:

```
@{f4}~{alt gd}wires.all{enter}{repeat}{down}{pause 2}
```

Notes of Caution for Creating Macros

When creating macros—whether or not they are repeating macros—care should be taken. First, certain commands should never be used in macros.



Do not use the Duplicate or Kill commands in repeating macros. Doing so raises the risk of accidentally killing the wrong stories or filling up the database and causing a “low-on-space” condition.

Secondly, all macros should be created using steps only to a point where varying options are not a possibility. For instance, a system administrator wants a macro designed to open a story and type a specific production cue on a certain line, then save the story and open the next one in the lineup. The system administrator must keep in mind that the process for editing stories may vary for each user depending on user preferences or queue location. In other words, in some cases, a user may start to save a story and be prompted by a dialog box that requests confirmation. This confirmation box is a user preference that varies with each user. A similar confirmation dialog box may appear if a user is opening a story in a read-only queue. So, any macro must incorporate these possibilities or stop prior to them. Otherwise, the macro may hang up on an unexpected dialog box.



To immediately stop a macro that is in progress for any reason—including one hung up on an unexpected dialog box—press the Escape (Esc) key.

If a macro should wait for a dialog box, such as Local Printing, before continuing, the syntax is: `{window Local Printing}`. The text following the word window must match exactly the title of the dialog box. If it does not match, the user must press the Escape key to exit the macro.

Keyboard Checker

Whenever you modify and save macros in a keyboard description story, the system checks the keyboard and all its macros for problems that may prevent the macro from working properly. If it finds a problem, it sends you a message describing the error.



Error messages are prefixed with `inws macro #%d`. The `%d` represents the macro number. Macros are numbered from one (1) at the beginning of the story. Some of the error messages will include the macro key identifier.

The iNEWS system will also issue a warning if any predefined system function keys—that is, those keys reserved for iNEWS system functions—are replaced with a macro. This warning can be ignored if you want to override the pre-defined system function keys with a macro. See “[Predefined Function Keys](#)” on [page 186](#) for more information.

7 Keyboards and Macros

As long as you get a `Keyboard ok` message, the description story can be used, but go back and fix any noted problems so the keyboard does what it is supposed to.

Error Messages

The following table contains a list of messages from the message dictionary that can appear after you save a keyboard description story.

| Error Message | Explanation |
|--|---|
| Duplicate key description (M_KEYDUP) | You defined the same function key twice in the story; remove one of the definitions. |
| First key description does not begin with @ (M_KEYSTART) | You must use an @ symbol as the first noncomment character in the description story. |
| Invalid key number (M_KEYRANGE) | You tried to define a function key with a number that is not supported. |
| Keyboard description contains too many characters (M_KEYLONG) | This description story is too long. Shorten the macros by using command abbreviations, or deleting macros you do not use. |
| Keyboard NOT usable (M_KEYBAD) | You must fix the errors in this description story before you can use it. |
| Keyboard ok (M_KEYOK) | You may use this keyboard, even if it has errors. |
| Missing key number separator (~) (M_KEYSEP) | You must follow the key number of an extended programmable key with a tilde (~). |
| Not enough key descriptions (M_KEYMIN) | You must include lines for all the standard VT keys. Include only the key's number on a blank line, if you don't want to assign a function to a standard key. |
| Warning: a key definition contains a repeating function (M_KEYREP) | One of your key definitions has a repeating function. If this is not what you want, edit the description. |
| Warning: badly placed @ exists in key definition line (M_KEYFUNKY) | A line in the story contains an @ symbol that is not the first character in the line or between two commands. If this is not what you want, edit the description. |
| Memory allocation error(M_BADMEMORY) | An internal error; contact Avid. |
| Unable to stack keywords (M_BADSTACK) | An internal error; contact Avid. |
| Mismatched {} (M_MISMATCH) | The macro does not have paired brackets. |

| Error Message | Explanation (Continued) |
|---|---|
| No { } found for reserved word (M_RESWORD) | A reserved word (key name) was found but was not enclosed in brackets. |
| Could not locate }%d (M_NOLOCATE) | Unbalanced { } pairs. Missing a }. The %d represents the occurrence number of the expected }. |
| Circular reference to macro #%d: (M_REFERENCE) | A circular reference was found in the macro. The %d represents the macro number of the reference. |
| Multiple tildes (~) found (M_TWOTILDES) | A macro definition cannot contain more than one tilde (~). |
| Multiple macro keys: %s %s (M_TWOTAGS) | An invalid key indicator specified—it contains more than one key name. Each %s represents a key name. |
| Unexpected: %s (M_UNEXPECTED) | An invalid key indicator specified—it contains plain text. The %s represents the unexpected text. |
| Unknown macro key identifier: %s (M_UNKNOWN) | An invalid key indicator specified—it contains a key name that is not one of the recognized key names. The %s represents the unknown key name. |
| No tilde (~) found (M_NOTILDE) | There is no tilde (~) separating the key indicator from the key action. |
| No macro key tag (M_NOTAG) | There was no key name included in the key indicator. |
| Shifted numeric keypad 0-9 keys cannot be assigned macros (M_NUMKEYNOSHIFT) | You cannot assign a macro to a keystroke combination of Shift and a numeric keypad key. |
| Empty macro (M_EMPTY) | There was no key action data in the macro definition. |
| Duplicate macro definition: %s (M_TWODEFS) | There was a duplicate key indicator defined. The %s represents the key indicator. |
| Invalid pause interval (M_BADPAUSE) | The interval included in the pause command was not a number between one and sixty. |
| Isolated keyboard state: (M_LONESTATE) | A state key (Alt, Shift, or Ctrl) was specified but did not modify the action of any other key, such as: {shift} instead of {shift abc} |
| %s does not distribute (M_DISTRIBUTE) | A non-state key (not Alt, Shift, or Ctrl) was found while another key was still depressed, such as: {insert{home}}}. The %s represents the key still depressed. |

7 Keyboards and Macros

| Error Message | Explanation (Continued) |
|---------------------------|--|
| Ignoring: %s (M_IGNORING) | The state of a macro definition was expected but did not begin with an at character (@). All of the text is being ignored. The %s represents the text being ignored. |

Warning Messages

The following table lists the messages that appear if the keys reserved for iNEWS system or Windows-based functions are redefined.

| Reserved Key | Warning Message |
|---------------------|---|
| F1 | Warning: “Help” key redefined (M_STDHELP) |
| F3 | Warning: “Find Next” key redefined (M_STDFINDNEXT) |
| Alt+F4 | Warning: “Exit” key redefined (M_STDEXIT) |
| Ctrl+F4 | Warning: “Window Close” key redefined (M_STDCLOSE) |
| Ctrl+F5 | Warning: “Discard Changes” key redefined (M_STDDISCARD) |
| F5 | Warning: “Refresh” key redefined (M_STDREFRESH) |
| F6 | Warning: “Script Swap” key redefined (M_STDSSCRIPT) |
| F7 | Warning: “GoTo Priority Queue” key redefined (M_STDPRIORITYQUEUE) |
| Shift+F7 | Warning: “GoTo Alerts History” key redefined (M_STDALERTSHISTORY) |
| F8 | Warning: “Communicate Message Bar” key redefined (M_STDMESSAGEBAR) |
| Shift+F8 | Warning: “Communicate Message Show History” key redefined (M_STDMESSAGEHISTORY) |
| F9 | Warning: “Communicate Open/Close Mail key redefined (M_STDMAIL) |

Testing the Keyboard

After you create a keyboard story, you should always test the macros in it to make certain that it works the way you intended.

To test the keyboard:

1. After you finish a macro, save the keyboard.
2. Verify that the keyboard checker lists no errors.
3. Log out of iNEWS and log back in; otherwise, the keyboard will not be available.
4. Select Tools > Options > Preferences to assign the keyboard and load it.
5. Test the macro(s) you created to make sure they perform as intended.

Assigning a Default Keyboard to a User Account

Users can select a keyboard—that is, a set of macros—to use at any time by using the Preferences option in the Tools menu. The system administrator can assign the default keyboard for a user, which appears when the user first logs on.

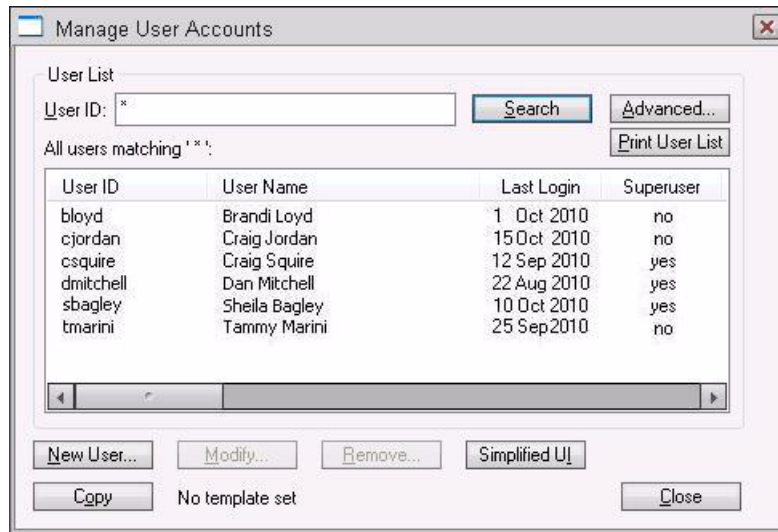
When you add a new user to the system, you may want to assign a keyboard as a default for the user at that time, according to the role the user plays in your newsroom. For instance, a new writer may get the “user” keyboard, a new producer would be assigned the “producer” keyboard, and so forth.

You assign a set of macros as a user’s keyboard by assigning the keyboard story containing those macros to that user. When the user presses a programmable key, the system looks at the user’s assigned keyboard story and executes the macro assigned to that key.

7 Keyboards and Macros

To assign a keyboard as a user's default from an iNEWS Workstation:

1. Select Tools > Options > Users. The Manage User Accounts dialog box appears.



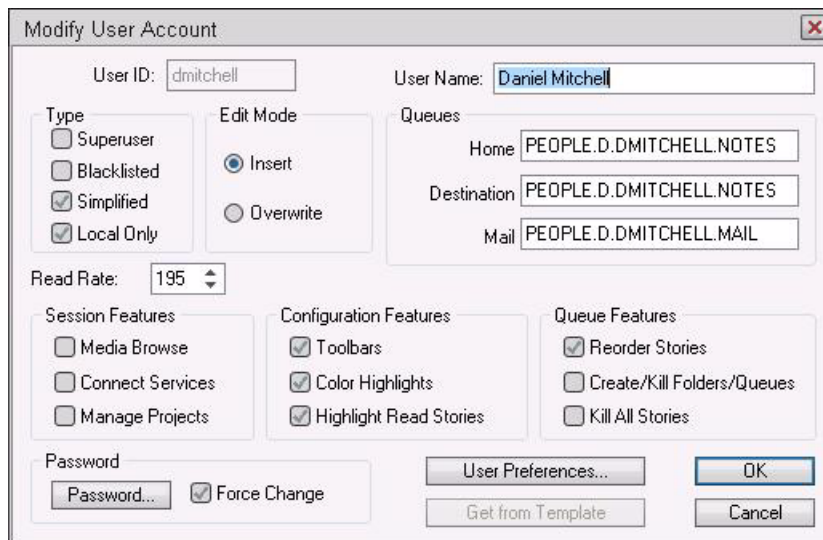
2. Do one of the following:

- ▶ Create a new user account by clicking New User.
- ▶ Modify an existing user account by selecting a user and clicking Modify.



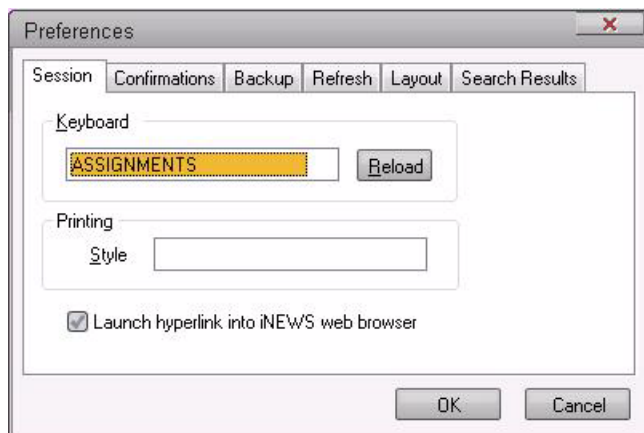
If you clicked New User rather than Modify in this step, the dialog box will be titled Add New User instead of Modify User Account. See “Adding a New User Account” on page 88 for more information.

3. In the Modify User Account dialog box, click User Preferences.



Access to the Modify User Account or Add New User dialog boxes is restricted to certain users, such as system administrators and user managers—that is, users who know the umanager password. See “Modifying User Traits” on page 68 for more information.

4. In the Preferences dialog box, click the Session tab, if not already selected.



5. Use the Keyboard drop-down list to select a keyboard for the user.
6. Click OK to save the new keyboard assignment and close the Preferences dialog box.
7. Click OK to save the modified user account and close the Modify User Account dialog box.

7 Keyboards and Macros

8. Do one of the following:
 - ▶ Select another user to assign a keyboard to, and click Modify.
 - ▶ Click Close to close the Manage User Accounts dialog box.
9. The next time the user logs in, the computer will automatically assign the new keyboard. If the user is already logged in, he or she needs to reload the keyboard or log out and then log back in to use the new keyboard assignment.



Users can choose a different keyboard than what is assigned to their user accounts at any time by doing the following: Select Tools > Options > Preferences. Then, choose the Session tab, pick a keyboard from the drop-down list, and click the Reload button.

8 Forms

You can use standard iNEWS forms that come with the system or customize your database by creating forms and assigning them to queues, based on the kind of information you want to appear in the stories in those queues.

This chapter contains the following main sections:

- [Form Guidelines](#)
- [Creating Forms](#)
- [Customizing Forms](#)
- [Assigning a Form as a Queue or Story Form](#)
- [Form Field Types and Definitions](#)
- [Standard iNEWS Forms](#)

Form Guidelines

You create forms that are stored as stories in queues located in the database file structure under the main system folder, SYSTEM.FORMS. The format is SYSTEM.FORMS.N.NAME, where N is the first letter of the form name.

For instance, a story form may be SYSTEM.FORMS.S.STORYFORM. A rundown form may be SYSTEM.FORMS.R.RUNDOWN. You can have up to 250 forms starting with each letter.

Follow these general guidelines in designing forms:

- The story form and queue form can be different, but all fields displayed in the queue form must exist in the story form before you can enter or display data in that field.
- In iNEWS, if you go over a page width, the overflow will be printed on a second page. Columns do not break across page boundaries.

Creating Forms

You can create a new form from any iNEWS Workstation. The following procedure takes you through steps to create the basic form for a story, which can later be used to create any other form in the system, such as a rundown form. After building the story form, you can easily copy and modify it to match the rundown you currently use.

To create a new form:

1. Navigate to the SYSTEM.FORMS folder. The queue for the new form must be stored in the SYSTEM.FORMS folder.
2. Select Tools > New Folder. A new highlighted directory labeled New-Folder appears.

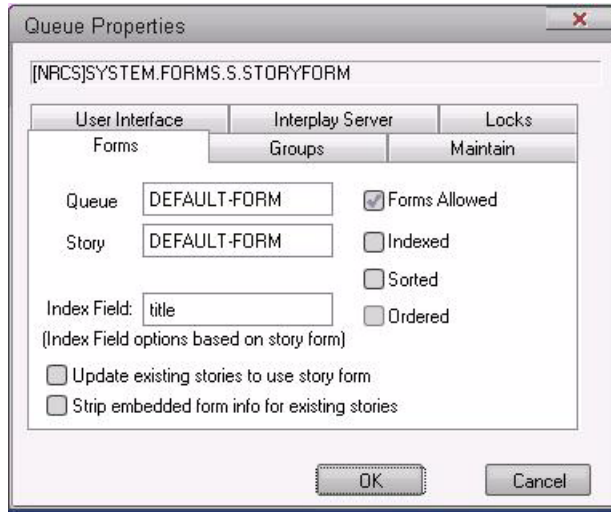


You can skip steps 3 & 4 if the alphabet folders already exist in the SYSTEM.FORMS directory.

3. Type the name of the folder, such as S.
4. Select the alphabet folder, such as S (the folder for forms with names that begin with the letter S), in which you want to create the new form.
5. Do one of the following:
 - ▶ Select Tools > New Queue.
 - ▶ Right-click on the folder and select New Queue from the context menu.

A new highlighted file labeled New-Queue appears.

6. Type the new form queue's name, such as STORYFORM.
7. The Forms Allowed attribute should be inherited from the parent queue, SYSTEM.FORMS, but you can ensure the Forms Allowed database trait is applied to the queue by doing the following:
 - a. Right-click on the queue in the Directory panel to open the Queue Properties dialog box.
 - b. Select the Forms Allowed check box so that a check mark appears.
See [“Changing Database Traits” on page 136](#) and [“Forms Tab” on page 123](#) for more information.
 - c. Click OK to save changes. Otherwise, you will be unable to create a form in that queue.



8. Double-click the queue to display it in the Queue panel.
9. Do one of the following:
 - ▶ Select File > New Story.
 - ▶ Position your cursor in the Queue panel and press the Insert key.



If the story form fields do not appear, click on the Story drop-down menu and select the option to Show Form Area.

8 Forms

A new story row appears in the Queue panel and opens in the Story panel. At the top of the Story panel, there is the Story Form panel that displays the story form fields (using the default form or the form previously assigned to the queue). The following graphic shows the Queue panel (on top with a new row) and the Story Form panel (on bottom with six standard form fields).

| Slug | Created | Modified | By | ? | TIME |
|------|------------------|------------------|-----|-------|------|
| | 2/22/00 12:29:25 | 2/22/00 12:29:32 | nzr | READY | |

| Slug | Created | Modified | By | ? | TIME |
|----------------------|------------------|------------------|-----|-------|------|
| <input type="text"/> | 2/22/00 12:29:25 | 2/22/00 12:29:32 | nzr | READY | |

10. Modify each field to customize the new form. For instance, you would need to select each field's type, which defines the field's function within the form. See [“Customizing Forms” on page 201](#) and [“Form Field Types and Definitions” on page 208](#) for more information.

Since a basic story form does not necessarily use the default fields, as shown in the above graphic, field modification is needed. For instance, the form may include the following fields:

- Page field uses the PAGE-NUMBER field type.
- Title (or Slug) field uses the TITLE field type.
- Presenter (or Anchor) field uses the PRESENTER field type.
- Writer (or Reporter) field uses either the CREATE-BY or WRITER field type.



When a new story is created, and the field called Writer is a CREATE-BY field type, it is filled in automatically but cannot be changed by users. However, if the WRITER field type is selected for the field called Writer, then when a new story is created, the field is filled automatically, but data in it can be changed manually by a user.

- Graphics (or Production Notes) field uses the VAR-N field type, where N is any number. Common practice is to name these fields v-*<description>*, such as v-graphic or v-notes.
- Read time field uses the AUDIO-TIME field type.
- Back time field uses the BACK-TIME field type.



Text and production cues may be placed into the body of the form and text may be put into the form fields; if so, all data contained in the story form will appear in a new story whenever one is created in a queue with that story form.

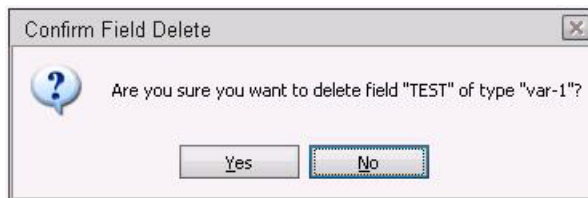
11. Save the story and exit the queue, accepting current fields and form properties as the new form.

Customizing Forms

After you create the form queue in SYSTEM.FORMS, you can modify fields to customize the new form.

To customize a form, open its story and do the following:

1. Put the cursor in the Story Form panel and right-click. A context menu will appear.
There are three menu options for customizing form fields: Insert Field, Delete Field, and Field Properties. Access to these options vary, based on whether or not you right-click on a field in the Story Form panel.
Another option in the context menu is Label Borders. See [“Turning on Label Borders” on page 206](#) for more information.
2. Choose an option based on one of the following: (The choice you make also determines which dialog box appears.)
 - ▶ Select Delete Field if you want to remove an existing field from the form. (The field you right-clicked on is the one you will delete if you choose this option.) The Confirm Field Delete dialog box will appear. Go to step 3.



- ▶ Select Field Properties to modify properties of an existing form field. The Form Properties dialog box appears. Go to step 4.
 - ▶ Select Insert Field to add a new field to the form. The Insert Field dialog box appears. Go to step 5.
3. Confirm your command to delete the field and, if necessary, return to step 2 to continue modifying form fields.

8 Forms

4. Use the Apply to radio buttons to determine whether you want your property modifications to be applied to the:
 - Current field—the one you right-clicked on
 - Current row—all fields in the row of the field you right-clicked on
 - Entire form—all fields in the form

The screenshot shows the 'Form Properties' dialog box with the following settings:

- Apply to:** Current field, Current row, Entire form
- Current field:** Label: PAGE, Label size: 5, Type: page-number, Edit size: 4, Starts new row
- Attributes:** Read-Only, Affects Ready, Write Group: [empty]
- Type:** Editbox, Checkbox, Calendar Control, Duration Control, Combobox, Allow Blank Selection, Editable
- Text alignment:** Left, Center, Right
- Label placement:** Top, Bottom, Left, Right
- Label alignment:** Left, Center, Right
- Text style:** Bold, Italic, Underline
- Label style:** Bold, Italic, Underline
- Entries:** [empty list], Add List/Group button
- Color palette:** No Color, Red, Green, Yellow, Purple, Pink, Orange, Grey, Blue, White





The rest of the Form Properties dialog box offers the same form options as those explained in step 5.

5. Select various form options to determine appearance and function of field(s) in the form.

The options available are explained in the following table.

| Option | Explanation |
|----------------|---|
| Label | Enter current field's name you want to appear in the Story Form panel. |
| Type | Select a field type from the drop-down list, which defines the field's function. This is different from the Type section on the right side of the dialog box, which is explained further in the next step of this procedure. See “Form Field Types and Definitions” on page 208 for a detailed explanation of the various types you can choose from the drop-down list, including variable fields that allow you to make up your own field names, such as “shot” or “printed.” |
| Starts new row | Select this check box to force the field to be the first one in the next row of the form. |
| Label size | Enter a numerical value to determine the label size. This size is in approximate characters. Using a proportional font will, of course, cause the number of characters to vary. Generally, label size should be set to edit size plus one. |

| Option | Explanation |
|-----------------|---|
| Edit size | <p>Enter a numerical value to determine the space text will fill in the field.</p> <p>The numerical value entered for edit-size does not limit the amount of text that can be typed in the field, but just the amount of text viewed in the field at any one time.</p> <p> <i>System administrators can set a limit for text in Story Form fields, using a Registry value defined as VT Compatibility at each workstation.</i></p> |
| Attributes | <p>Select attributes for the form field.</p> <ul style="list-style-type: none"> • Read-Only check box determines whether the form field can be read (not modified). • Affects Ready check box determines whether the form field participates in determining the Ready field value. • Enter a group name in the Write Group text box. When a write group is assigned to a form field, only members of that group and superusers can edit that field. |
| Text alignment | Select text position within the field. |
| Label placement | <p>Select label position.</p> <ul style="list-style-type: none"> • Top (default) puts label on top of the field • Bottom puts label below the field • Left or Right puts label to either side of the field. |
| Label alignment | Select way in which label aligns (left, center, or right) with the field. |
| Text style | Select appearance of text (bold, italic, or underline) in the field. |
| Label style | <p>Select appearance of the label (bold, italic, or underline).</p> <p> <i>Any combination of these attributes may be selected.</i></p> |

6. Specify the Type.

| Option | Explanation |
|---------|---|
| Editbox | Select the Editbox radio button to make the field a standard text-entry field. Any user-created field type can be assigned as an editbox field. |

| Option | Explanation |
|------------------|--|
| Checkbox | <p>Select the Checkbox radio button to make the field a checkbox. Any user-created field type can be assigned as a checkbox. Checkboxes will always have the Text alignment attribute set to Center. The values are "1" for checked and blank for unchecked, allowing use with raw queries and Affects-Ready. Set the value by selecting or deselecting the checkbox once the field is inserted into the form. Checkboxes can be selected with the mouse or with the spacebar.</p> |
| Calendar Control | <p>Select the Calendar Control radio button to make the field a calendar field in the form. Any user-created field type can be assigned as a calendar field. Data entry can be via keyboard, spinner, or the calendar control that appears when selecting the button in the field. The calendar field defaults to a time of 00:00:00, but times can be added to the dates if more specificity is required. Dates are stored on the server as seconds since January 1, 1970.</p> |
| Duration Control | <p>Select the Duration Control radio button to make a field a duration field. Any user-created field type can be assigned as a duration field. Duration fields allow keyboard- or spinner-based entry. They appear as HH:MM:SS. Durations are stored on the server as seconds.</p> |
| Combobox | <p>Select the Combobox radio button if you want to assign predefined values to the field. When these fields are used in a form, the user is presented with a drop-down list instead of a text field. You can define the values by using a system list or group, or you can create an ad-hoc list of values.</p> <ul style="list-style-type: none"> • Select Allow Blank Selection if you want a user to be able to choose a blank value to clear the field. • Select Editable if you want a user to be able to type their own value into the combobox as opposed to selecting from the predefined values you provide in the drop-down list. • Select the Add List/Group button to open a new dialog box so you can select system lists or groups to be used to populate the combobox. Your selection will appear as an entry in the Entries section. See “SYSTEM.LISTS” on page 207 for more information about lists. • Use the Entries section to create ad-hoc values that will appear in the combobox. In this section of the dialog box, you can use the four buttons to add new values, delete an existing value, or rearrange the order of your pre-defined values. • The color boxes show the current system colors. Select an entry and then a color to assign that color to the chosen entry. Colors can be assigned to individual values or to entire groups and lists. When a color is applied, a comma followed by a number (1-10) will appear after the entry to indicate which color index is assigned to the entry. See “SYSTEM.COLORS” on page 207 for more information about colors. |

7. Do one of the following:
 - ▶ Click OK in the Form Properties dialog box to record changes to an existing form and close the dialog box.
 - ▶ Click Insert Before or Insert After in the Insert Field dialog box, depending on where you want the new field to appear in the form in relation to the field (or cursor position) you right-clicked on in step 1. This will record the changes and close the dialog box.
8. Save story in the form queue, by clicking the Save button or selecting File > Save.

After a form is created and customized, it can be assigned to other queues in the database as either the queue form, which dictates appearance of the queue, or story form, which dictates appearance of stories created in the queue.

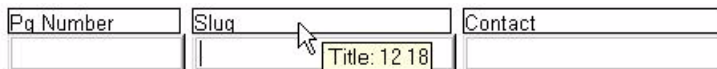
Turning on Label Borders

Label borders provide various information about fields in a tool tip format.

To turn on label borders in the Story Form panel:

1. Right-click in the Story Form panel.
2. Select Label Borders in the context menu.
3. Position mouse pointer over a field label to view tool tip.

When selected, rectangular borders are placed around the labels for each field in the form. When the mouse pointer is positioned over a field's label, a tool tip appears that displays that field's type, and the character count for the field position and label size.



For instance, the field called Slug (shown above) is a TITLE field located 12 characters in from the left with a label size of 18 characters. So, from the position of the Slug field, you can determine the Pg Number field has a Label size of 12 characters. Also, the Contact field, in the example, is located 30 characters in from the left—that is, its position in the row is equal to the sum of the label sizes for the fields directly to its left. This is the case for sites that follow the general recommendation by Avid that label sizes be equal to each field's edit size plus one. However, if the fields' edit sizes are larger than their label sizes, a field's position in the row would be based on the larger of the two. You may confirm these figures by positioning your mouse over the fields and viewing the tool tips that appear for each field. Similar to those that appear for labels, field tool tips also provide character count for position and edit size.



Positioning labels to the left or right of fields rather than above or below will also result in adjustments for calculating field positions on a row, since both the label and edit sizes would have to be included instead of the larger of the two.

SYSTEM.COLORS

There is a queue that holds system colors, which can be chosen (in the Form Properties or Insert Field dialog boxes) to customize form fields. By default, the colors reside in the queue SYSTEM.COLORS as defined by the Q_COLORS token in /site/dict/queues.

The first story in SYSTEM.COLORS defines the system's colors. The format for each color definition in the story is:

```
<index> <red RGB value> <green RGB value> <blue RGB value> ;<comment>
```

Only indexes 1 through 10 are accepted, and the color values are numerical from 000 to 255. For instance, the first three color definitions in a story might appear similar to:

```
1 255 000 000 ;red
2 000 255 000 ; green
3 255 255 000 ; yellow
```

SYSTEM.LISTS

As a system administrator, you can create system lists, which can be chosen (in the Form Properties or Insert Field dialog boxes) to customize predefined drop-down lists in comboboxes. By default, the lists reside as stories in queues located in the directory, SYSTEM.LISTS, as defined by the Q_LISTS token in /site/dict/queues.

The SYSTEM.LISTS directory is organized in a structure similar to SYSTEM.FORMS, where the list is the first story in a queue that bears the same name as the list, and the first character of that name is the name of the sub-folder in which the list's queue and story are located. For instance, a list called format is the first story in the queue SYSTEM.FORMS.F.FORMAT.

The format of a list story is: <text value> <background color index> ; <comment>

Each text value is a predefined list entry that will appear as an item in a combobox's drop-down list. Double quotes should be used if special characters or spaces are used in the text value. Assigning a background color is optional, and if left blank, no color is used. Here is an example of a list:

Stop 1 ; the Stop value has a red background color

Go 2 ; the Go value has a green background color

"No Light" ; the No Light value has no background color

Assigning a Form as a Queue or Story Form

Queue and Story forms are database traits that allow you to assign different forms to different folders and queues using the Directory/Queue Properties dialog box at the workstation, or the dbtraits command at the console.

You assign these traits to define the appearance of information in the Queue panel and Story Form panel.

To assign a form at an iNEWS Workstation:

1. Navigate to the directory (folder) or queue you want in the Directory panel.
2. Right-click to open the Directory/Queue Properties dialog box. Access to this dialog box and its appearance varies, depending on certain circumstances. See ["Directory/Queue Properties Dialog Box" on page 122](#) for more information.
3. Do either or both of the following:
 - ▶ Use the Queue drop-down list on the Forms tab to select the form you want to apply to the directory as queue form database trait.
 - ▶ Use the Story drop-down list on the Forms tab to select the form you want to apply to the directory as story form database trait.See ["Forms Tab" on page 123](#) for more information.
4. Click OK to save changes and apply the new queue/story form settings.



Users should log off and sign back on to view the new queue/story form settings.

Form Field Types and Definitions

Form field types are explained in this section. Included is the suggested maximum or minimum length for each field (where applicable) and whether a user can enter text in the field.

You can reuse only the VAR-*N* and AFF-READY-*N* fields in a form, where *N* is a different number for each occurrence, such as AFF-READY-1. You can use all other field types only once.



The following table provides explanation of forms pertaining to broadcast and/or machine control and which fields are typically used in a variety of forms/queues.

| Field Type | Description |
|---|--|
| AFF-READY- <i>N</i> (<i>N</i> represents any number.) | AFF-READY- <i>N</i> fields are created when a system is converted from Avid Netstation to iNEWS. The AFF-READY- <i>N</i> fields are assigned the “Affects Ready” attribute when created. This means the field affects the display in the READY status field and allows a user to initialize a story and change its status. There can be more than one AFF-READY- <i>N</i> field in a form. If any AFF-READY- <i>N</i> field within a form contains a “?” then the READY field displays a NOT READY message. If a question mark does not appear in any of the AFF-READY- <i>N</i> fields in the form, then the Ready field displays a READY message. This field should precede the READY status field in the form. In iNEWS, the “Affects Ready” attribute can be assigned to any field in a form and the result would be the same behavior as described here pertaining to the AFF-READY- <i>N</i> field. For more information, see the definition for the READY field in this document. |
| AIR-DATE | When a story is aired using the show-timing function, the date and time it airs is inserted in this field. This field is designed to show which story is currently on-air and give users a sense of how much time remains until a later story goes to air. As the producer syncs timing on stories, the queue display on other iNEWS Workstations shows that story in a different color. (Peach is the standard default color rule.) The format on iNEWS Workstation is controlled by the workstation control panel for regional settings. |
| APPN- <i>X</i> to APPN- <i>X</i> (<i>N</i> represents 1 through 5; <i>X</i> represents any number.) | These fields are used differently, depending on the queue in which they are used. They appear in Machine Control Terminal (MCT) forms and some Command or ControlAir forms. Also, many of the fields have different meaning, depending on the device type. For instance, in an MCT form, the event ID for a video/cart machine goes in the APP1-1 field, but in the APP2-1 field for a CG or SS, where the APP1-1 field is used for the style. These fields were previously used in the mail and account forms in Avid Netstation, but are no longer used for those forms in iNEWS. The APPN- <i>X</i> fields have been renamed in iNEWS and occur only in template conversion from Avid Netstation. |

| Field Type | Description (Continued) |
|--------------|--|
| AUDIO-TIME | <p>This field will display the estimated time for reading a story, which can be estimated by the computer system or entered into the field by a user. If time is in the field, the system will use it to calculate the total time. If there is also a TAPE-TIME field in the form, the system adds the TAPE-TIME to the AUDIO-TIME to calculate the story's total time.</p> <p>Without any user input, the system will display an estimated time based on the length of the story and presenter's read rate, which is obtained from the PRESENTER field in the form. If there is no PRESENTER field or it does not contain a user ID or the ID in the field does not have a read rate associated with it, then audio time is based on the system's default read rate. The length of the story is actually the word count of the story, since the read rate is based on words per minute.</p> <p>If a user has entered a time in the field and wants to restore the audio time calculated by the system, the user should remove entered data from the field with the space bar, delete, or backspace key. After the cursor leaves the form field, the system will then display the computer-calculated audio time and recalculate total time accordingly.</p> |
| BACK-TIME | <p>The system displays the back-time in this field. The back-time field is usually eight characters wide, such as 00:00:00. A user can enter data in this field to indicate hard-hit times for back-timing to certain points within a program.</p> |
| CA-CAPTURED | <p>This field displays total number of characters captured during a session connection; it is one of eight special fields used in the SYSTEM.ACCOUNT queue form for logging connection time activity. Although this field is not required, omitting it will prevent iNEWS from displaying the corresponding information in the form.</p> |
| CA-DIRECTION | <p>This field displays the direction of incoming or outgoing connections. It is one of eight special fields used in the SYSTEM.ACCOUNT queue form for logging connection time activity. Although this field is not required, omitting it will prevent iNEWS from displaying the corresponding information in the form.</p> |
| CA-ELAPSED | <p>This field displays elapsed time of a session connection. It is one of eight fields used in the SYSTEM.ACCOUNT queue for logging connection time activity. Although this field is not required, omitting it will prevent iNEWS from displaying corresponding information in the form.</p> |
| CA-IDENT | <p>This field displays the connection identifier. It is one of eight fields used in the SYSTEM.ACCOUNT queue for logging connection time activity. Although this field is not required, omitting it will prevent iNEWS from displaying corresponding information in the form.</p> |


| Field Type | Description (Continued) |
|-------------|---|
| CA-ORIGIN | This field displays the origin machine name. It is one of eight special fields used in the SYSTEM.ACCOUNT queue form for logging connection time activity. Although this field is not required, omitting it will prevent iNEWS from displaying corresponding information in the form. |
| CA-RECEIVED | This field displays the total number of characters received from a remote system during a connection. It is one of eight special fields used in the SYSTEM.ACCOUNT queue form for logging connection time activity. Although this field is not required, omitting it will prevent The iNEWS system from displaying corresponding information in the form. |
| CA-REMOTE | This field displays the remotely connected machine name. It is one of eight special fields used in the SYSTEM.ACCOUNT queue form for logging connection time activity. Although this field is not required, omitting it will prevent iNEWS from displaying corresponding information in the form. |
| CA-SENT | This field displays the total number of characters sent to a remote system during a connection. It is one of eight special fields used in the SYSTEM.ACCOUNT queue form for logging connection time activity. Although this field is not required, omitting it will prevent iNEWS from displaying corresponding information in the form. |
| CG-ADDR | This field holds the ID's or recorded page addresses from the character generator on which a super is written by the CG interface. This field is primarily used in association with ControlAir systems. See “iNEWS Control Air Fields and Forms” on page 218 for further information on when and how this field is used. |
| CG-TEMPLATE | This field contains template information for the character generator, namely the address on the character generator of the template or tab field to be used for the requested super. This field is primarily used in association with iNEWS Command or ControlAir systems. See “iNEWS Control Air Fields and Forms” on page 218 for further information on when and how this field is used. |
| CG-TEXT | This field contains text of the super from the machine control instruction requested by a user in the script. It is written into specified template fields on the character generator interfaced with iNEWS. This field is primarily used in association with iNEWS Command or ControlAir systems. See “iNEWS Control Air Fields and Forms” on page 218 for further information on when and how this field is used. |

| Field Type | Description (Continued) |
|-------------|--|
| CHANNEL | The letter or numerical identifier of the on-air output or playback channel of an interfaced production device is located in this field. Most production devices have two or more channels. This field is primarily used in association with iNEWS Command or ControlAir systems. See “iNEWS Control Air Fields and Forms” on page 218 for further information on when and how this field is used. |
| CREATE-BY | When you open a new story, the system enters your user name in this field. The name is permanent and cannot be erased or overwritten. In a Mail form, this field is used to indicate who sent the e-mail. |
| CREATE-DATE | When you create a new story, the system stores the date and time the story was created in this read-only field. The format on iNEWS is controlled by the workstation control panel for regional settings. |
| CUME-TIME | The computer displays the cumulative (cume) time. This field represents the time of all the stories—except the selected story—added together. It has a suggested minimum length of five characters, and is typically eight characters. |
| DEVICE-MGR | This field displays the device name controlling a particular event. This field is primarily used in association with ControlAir systems. See “iNEWS Control Air Fields and Forms” on page 218 for further information on when and how this field is used. |
| DURATION | This field can be used to store manually entered, suggested duration. The value is not used for time calculation in any other fields. |
| EFFECT | This field holds the effect name requested in association with a machine control or broadcast control event that will be applied to the character generator or still store machine when it’s taken to air. For instance: a wipe or a dissolve. This field is primarily used in association with ControlAir systems. See “iNEWS Control Air Fields and Forms” on page 218 for further information on when and how this field is used. |


| Field Type | Description (Continued) |
|--------------|---|
| ENDORSE-BY | <p>This field enables a user with write-access to the queue to endorse stories in that queue. When a story is first saved in a queue, the field is red and blank. Whenever a user endorses the story, the system places that user's name in the ENDORSE-BY field corresponding to that story and the field changes to green.</p> <p>To manually endorse a story, click on the field in the queue. If another user subsequently changes the story and saves it, the ENDORSE-BY field turns yellow but the endorser's name remains in the field. This indicates the story was changed by one user after it had been approved by another. The endorser can see who made the change by looking in the MODIFY-BY field if there is one in the story form. The endorser can also withdraw approval by opening the story form and deleting the user id from the ENDORSE-BY field.</p> <p>If the ENDORSE-BY field is not shown in the Queue panel, a user can still endorse the story by typing a character in the ENDORSE-BY field located in the story form. A non-system administrator (non-superuser) cannot kill an endorsed story if another user has the Production Lock set.</p> <p> <i>The story form must include a MODIFY-BY field to show the green "endorsed" status in the ENDORSE-BY field.</i></p> |
| EVENT-STATUS | <p>This field displays availability and play status of a ControlAir, Command, or MOS event, as reported by the production device involved. For instance, a video event could be reported as OFFLINE, CUED,PLAYING, or STOPPED, among other things. In rundown and Event List forms only the status of a video event can be displayed. In ControlAir Workstation forms, this field can also contain the status of CG and still store events. See “iNEWS Control Air Fields and Forms” on page 218 for further information on when and how this field is used.</p> |
| ITEM-CHANNEL | <p>This field is required to allow changing of the channel that primary events will play on. It does not apply to events entered as production cues in the story body. The ITEM-CHANNEL field is used to set or display the channel for primary Command, MOS, or ControlAir video events.</p> <p>A user can edit this field—thereby changing the play out channel for a primary event—manually or through the Assign Channel dialog box at the iNEWS Workstation. Any changes made at an iNEWS Workstation will be relayed to the appropriate device.</p> <p> <i>When using the Assign Channel dialog box for MOS plug-in events, the channel information is displayed in the ITEM-CHANNEL field.</i></p> |
| LITERAL | <p>This field is a non-editable field, typically used as a label or spacer to assist in alignment of other fields within the Story Form.</p> |

| Field Type | Description (Continued) |
|-------------------|---|
| MAIL-CC | This field is used in the Mail story form to display names of users receiving a copy of an e-mail message. |
| MAIL-TO | This field is used in the Mail story form to display names of users to whom an e-mail message is sent. |
| MODIFY-BY | This field contains the name of whoever most recently modified a story. A user cannot edit this field. |
| MODIFY-DATE | Displays date and time story was last modified. Every time a story is edited and saved, the system updates this field. |
| MODIFY-DEV | If a device name for a workstation is included in the configuration file, it appears in this field when a story is saved at that workstation. If the device name is not included in the configuration file, this field displays the IP address of the workstation that was used when the story was saved. The maximum number of characters for a device name is eight, but the IP address can contain up to 15 characters. |
| MOS-ACTIVE | This write-protected field is required for the Story Form to accept MOS or CAP (ControlAir Plug-in) events. The field's content is created by the software when MOS or CAP events are created. Content may be cleared if the user selects Delete Machine Control from the right-click pop-up menu. If the VIDEO-ID field is also present, it will be read only whenever there is a value in MOS-ACTIVE. |
| MOS-DURATION | <p>This editable field is required to allow duration information from MOS or CAP events to be included in calculations for CUME-TIME, BACK-TIME or TOTAL-TIME fields.</p> <p>A user can enter a value, which appears bold, directly into this field, which will be used only in show-timing calculations. Such user-entered information does not become part of MOS or CAP events that are passed on to the appropriate MOS device or ControlAir in the case of CAP events. Deleting user-entered information from the field will allow show-timing to revert back to duration information supplied by MOS or CAP events.</p> <p>The contents of MOS-DURATION field are only used if a RUNS-TIME field is present in the story. The contents of the MOS-DURATION field is added to the computed runs time of the story and this total is shown in the RUNS-TIME field. If the user enters a time in the RUNS-TIME field, the MOS-DURATION field contents are ignored.</p> |
| MOS-SUBEVENT | This field is reserved for future use. |
| MOS-TITLE | This write-protected field is required to allow the display of descriptive text associated with MOS or CAP events. |

| Field Type | Description (Continued) |
|--------------|---|
| NSML-LITERAL | This field is only seen in stories and forms transferred from an Avid Netstation database to an iNEWS database and cannot be created by a user. The field is a by-product of the database conversion process and represents a protected field in the Netstation template. You can delete it and replace it with a usable field. |
| PAGE-NUMBER | This field is used primarily to arrange stories in a queue. If a queue is set to sort stories by page number, users can change the order of stories in the queue by entering in new page numbers. This is commonly done with rundowns, because it allows the producer to quickly change the order of stories by assigning them new page numbers. This field can also be displayed on a myriad of external systems, such as Command, ControlAir, or some third-party prompters. The system uses the first six characters of the field as a page number when the story is printed. This field is used in printing only when the story is printed using the “print script” option. |
| PRESENTER | The system uses the name of the user from this field to look up the user read-rate. The system then calculates audio time for the story based on the user read-rate. If a form does not contain a PRESENTER field, the system calculates the audio time, based on the default read-rate set in the system profile. (This is usually 180 wpm in English.) |
| READY | <p>In some cases, the system places status information, such as NEW, HOLD, LOCKED, or WIRE in this field. For instance, a wire story starts with the WIRE status. When a user changes the story, it changes from WIRE to READY. In the Mail form, the READY field is used to indicate whether an e-mail message is NEW or was READ.</p> <p>A user cannot enter data in this field, but it can be altered based on information put in other fields with the “Affects Ready” attribute in the form. In these cases, iNEWS checks the “Affects Ready” fields in the form and if any one of the fields are empty or contain a question mark (?) as the first non-blank character, then the READY field displays a “?” status. However, iNEWS displays READY in the READY field if none of the fields with the “Affect Ready” attribute contain a question mark, and all Affects Ready fields are populated.</p> |
| RESULT-INDEX | This field is in forms used to define the display of iNEWS database search results. It contains the sequence number that indicates the original order of items within the search results. |
| RESULT-LOC | This field is in forms used to define the display of iNEWS database search results. It contains the location of data found during a search, such as the name of a queue containing the story found matching the search criteria. |

| Field Type | Description (Continued) |
|------------|---|
| RUNS-TIME | <p>This field displays the sum of all “runs=” times in a story’s production cues plus the contents of the MOS-DURATION field. If the RUNS-TIME field is not present in the story form, then the runs time entries will not be calculated. If it’s not present along with MOS-DURATION, then the time from the MOS-DURATION field will not be added to the story’s total time.</p> <p>Runs time entries and the RUNS-TIME form field are not updated when the iNEWS Workstation is connected only to the local database. When connected to both the local database and server, the runs times will be updated when the local database story is edited. Also, when the story is copied from the local database into a queue on the iNEWS Server, its runs times will then be updated accordingly.</p> <p>The field functions much like the AUDIO-TIME field in that it can accept user entered times while still maintaining the “real” runs time. For example, a user can manually enter a time of 45 seconds in the field and that time will be reflected in the TOTAL-TIME field. The sum of runs time entries as calculated in the RUNS-TIME field are also included along with time from story text in the text timing clocks, located at the bottom of the Story panel. However, if a user manually edit the RUNS-TIME field, that data will not be included in the times calculated for the text timing clocks.</p> <p>By default, the Runs Time feature uses the word RUNS (not case-sensitive) to indicate a runs time entry, which is quite flexible. Some examples are:</p> <pre data-bbox="465 951 634 1170"> RUNS=25 RUNS=1:30 RUNS = 25 RUNS 1:30 TAPE RUNS 115 </pre> <p>You can change this default by redefining the token W_RUNS on the server in /site/dict/words.</p> <p> <i>A runs time entry of 115, as shown in the last example above, is the equivalent of one minute and fifty-five seconds (1:55). The number 115, without a colon, is interpreted as the number of seconds. So, 115 is not the same as 1:15.</i></p> |
| SEARCH-ID | <p>This field, which contains the request ID number, is used by the Find All and Fast Text Search features of iNEWS.</p> |

| Field Type | Description (Continued) |
|--------------|---|
| STATUS | This field is used in a rundown queue. It will display “OK” or “ERROR,” depending on the machine control event. The iNEWS monitor server utility program sets this field to indicate whether there are any errors in the production cues in stories of the rundown. In the SYSTEM.ACCOUNT queue form used for logging connection activity, the STATUS field is used to display the type of connection. |
| STILL-ID | In an Event List or Composite List queue, this field displays the alphanumeric identifier for a still store graphic. This field is primarily used in association with ControlAir systems. See “iNEWS Control Air Fields and Forms” on page 218 for further information on when and how this field is used. |
| STILL-PRESET | This field contains the number or letter designation of a predefined still store format. It is typically used in the form for the still store device event list, and is recognized by the ControlAir Workstations. This field is primarily used in association with machine control systems and ControlAir systems. See “iNEWS Control Air Fields and Forms” on page 218 for further information on when and how this field is used. |
| STYLE | This field contains the MCS/ControlAir style name specified when a user requests a CG or Still Store event in the production cue. It is typically an alpha or alphanumeric sequence that is a maximum of eight characters long. For instance, iNEWS translates a CG style into an address on the character generator at which a template is stored. That template is then used to build the requested super. For ControlAir, styles are defined in stories in the SYSTEM.RESOURCE queue. They define the details, such as CG template, number of fields, still preset or playback effect, that define an event. This field is primarily used in association with ControlAir systems. See “iNEWS Control Air Fields and Forms” on page 218 for further information on when and how this field is used. |
| TAPE-TIME | A user can enter the tape’s run time in this field. If there is an AUDIO-TIME field, the system adds tape time to audio time to calculate the story’s total time. |
| TITLE | This field is used to give each story a name. Occasionally referred to as the “Slug” field, it is the default quick “index field” for sorting stories in a queue. The first 20 characters of any field defined as the sort field of a queue is extracted and used as the “quick index” for the story. The TITLE field is usually selected as the “index field.” That means it is the field searched when a user conducts a Find or Find All search function and specifies a search of the index field. In a Mail form, this field is used to display the Subject of the E-mail message sent. In the SYSTEM.ACCOUNT queue form used for logging connection activity, the TITLE field is used to display the Service name. |

| Field Type | Description (Continued) |
|------------|---|
| TOTAL-TIME | The computer stores the total time for a story in this field, calculated based on the sum of information from the RUNS-TIME, TAPE-TIME and AUDIO-TIME fields. If the AUDIO-TIME field is not present, the computed audio time is used. The RUNS-TIME and TAPE-TIME fields must be present to be included in the total time. The story's total time is used when calculating back-time. If none of the three field types are present, the total story time is a computed audio time. |
| VAR-N | <p>This variable field was carried over from the Avid Netstation template conversion. It is typically used for generic editable text fields. Users can make up their own new fields using descriptive names, such as "crew" for a field that lists a reporter/photographer team assigned to cover a story. The user can employ anything for a field name with the following restrictions:</p> <ul style="list-style-type: none"> • The name must be 12 characters or less • The name must begin with a letter of the alphabet • It can include any letters (a-z), numbers (0-9), a dash (-) or a period (.) <p>The name is not case sensitive. User-created field names are treated the same as variable fields.</p> |
| VIDEO-ID | This field is used in the rundown queue to display the tape number or clip ID for video. It is also found in forms for the composite and video event lists. This field is primarily used in association with machine control and Command or ControlAir systems. |
| WRITER | <p>When someone creates a new story, the system automatically enters his or her user name in this field if it is present in the form. You can also enter text and change the name.</p> <p> <i>Distribution codes assigned to wire stories are placed in the WRITER field if it is present on the form used for wire stories. The codes would also be placed in a MODIFY-BY or CREATE-BY field if either are present in the story as well.</i></p> |



Write-protected fields may be edited if the third-party vendor supplies an editor interface. Otherwise, such fields can be modified only by deletion or by replacement. To delete, use the Delete Machine Control option from the right-click pop-up menu. To replace, use the Import From Plugin option or drag-and-drop a MOS or CAP event from any source.

iNEWS Control Air Fields and Forms

Certain field options within iNEWS are used in specialized forms, which deal with various devices that interface with the iNEWS newsroom computer system. These forms work in part with the iNEWS ControlAir system monitoring programs.

In certain cases, some fields in these forms are filled out by the system rather than users. The monitor utility program in iNEWS recognizes a set of descriptive field types in the Rundown, Event list, and Composite list forms.

Here's a brief description of these forms:

| Data Type | Rundown Forms |
|----------------------|---|
| RUNDOWN FORMS | This is the form used by stories in the rundown queue. The monitor program may extract text from some fields in this form and may put text in others. It also copies text from the rundown form to Event/Composite lists. Only event information for video devices is currently placed in the rundown story form. |
| EVENT LIST FORMS | The monitor server program places information about an event destined for a specific device in the form fields of the Event List story. It also copies rundown field text—for instance, title and page number—into matching fields in the Event List story form. |
| COMPOSITE LIST FORMS | The monitor server program puts information about events destined for all devices in the form fields of the Composite List story. It also copies rundown field text—for instance, the title and page number—into matching fields in the Composite List story form. |
| ControlAir FORMS | The function of the iNEWS ControlAir Workstation form controls which data will be displayed on the playback screen. Currently, the iNEWS ControlAir Workstation only extracts the PAGE-NUMBER from this form. Fields defined for use only by the iNEWS ControlAir Workstation are: DURATION, CHANNEL, and DEVICE-MGR. |



See *“Monitor Servers”* on page 381 for more information about the utility program.

Below is a chart of form fields filled in by the monitor server (utility) program:

| Data Type | Rundown Forms | Event List Forms | Composite List Forms |
|------------------|----------------------|-------------------------|-----------------------------|
| Error Status | STATUS | | |
| Video ID/Address | VIDEO-ID | VIDEO-ID | VIDEO-ID |
| Video Status | EVENT-STATUS | EVENT-STATUS | EVENT-STATUS |
| Video Duration | TAPE-TIME | | |
| CG Address | | CG-ADDR | CG-ADDR |
| CG Template | | CG-TEMPLATE | CG-TEMPLATE |

| Data Type | Rundown Forms | Event List Forms | Composite List Forms |
|------------------|----------------------|-------------------------|-----------------------------|
| CG Text | | CG-TEXT | CG-TEXT |

Standard iNEWS Forms

The iNEWS system contains a number of standard forms, each designed for use with one of the system's features. The forms your system provides are ready to use, but you can also customize them. You can add or remove fields from a form or change a field's title.

Account Queue Form

The system keeps a record of events, such as who logged into a connect session, the service used, and length of the connect session. This information is made available when the system creates a story in the SYSTEM.ACCOUNT queue and puts information in the story's form. A form designed for this must be assigned to SYSTEM.ACCOUNT, or your system cannot display the information. The standard account queue form is usually stored in the queue SYSTEM.FORMS.A.ACCOUNT.

The following table shows the fields used in the Account Queue form.

| Field Label | Field Type |
|--|-------------------|
| Service name | TITLE |
| User | CREATE-BY |
| Date and time | CREATE-DATE |
| Connection type | STATUS |
| Origin machine name | CA-ORIGIN |
| Incoming or outgoing connection | CA-DIRECTION |
| Remote machine name | CA-REMOTE |
| Elapsed time of session connection | CA-ELAPSED |
| Connection identifier | CA-IDENT |
| Total characters sent to remote system | CA-SENT |
| Total characters received from remote system | CA-RECEIVED |
| Total characters captured | CA-CAPTURED |

You are not required to include any of these fields in the account queue form. However, omitting a field prevents the system from displaying information associated with that field in the account queue.

Mail Form

The Mail feature has a form specially designed for mail stories. This form has fields for elements such as the subject of your message, name of message recipient, and people you want copied on the message. The mail form is stored in SYSTEM.FORMS.M.MAIL. The system includes a mail form, but you can specify a form for the mail feature to use by assigning the form to the SYSTEM.MAIL.OUT queue.

You can modify this form to suit your system needs. The following fields are used by the Mail queue.

| Field Label | Field Type |
|-------------|------------|
| To | MAIL-TO |
| Subject | TITLE |
| Cc | MAIL-CC |
| From | CREATE-BY |

Only the To field is required. A form that does not have the other fields is still usable.

Find All Form

Your system uses the form assigned to SYSTEM.SEEK, or whichever queue is specified for Q_SEEK in the system dictionary (/site/dict/queues) as the find all form. Since the system expects to find certain information in each of the form's fields, the form assigned to SYSTEM.SEEK must be designed for use with the find all command.

The follow table shows field types used in the find all form.

| Field Label | Field Type |
|---------------|------------|
| ID | SEARCH-ID |
| Status | READY |
| Search For | VAR-1 |
| Search Where | APP1-1 |
| Results Queue | TITLE |

| Field Label | Field Type |
|-------------|-------------|
| Search Type | APP2-1 |
| Max Hits | PAGE-NUMBER |
| Notify Hits | APP3-1 |
| Hits so Far | PRESENTER |
| By | CREATED-BY |
| Started At | CREATE-DATE |



The find all command must have certain information to perform a search. If you leave out a required field, the iNEWS Workstation displays a Missing form field message whenever someone tries to use the find all command.

Wire Story Form

To display information such as title, time moved, source, and average reading time in each wire, the system contains a wire story form. It is assigned to the wire queues and stored in SYSTEM.FORMS.W.WIRES. Any changes you make to this form do not take effect until you restart the wire. The following table shows field types used in the wire story form.

| Field Label | Field Type |
|-------------|---------------------|
| Title | TITLE |
| From | WRITER or CREATE-BY |
| Moved | CREATE-DATE |
| Status | READY |
| Time | AUDIO-TIME |
| FlashWord | PRESENTER |
| Number | PAGE-NUMBER |



The name of the form used for wire stories is defined by the Words dictionary token W_WIRE_FORM.

9

Character Generator Title Entry

The iNEWS newsroom computer system provides journalists, producers, directors, writers, and technical personnel in a newsroom with an array of tools to make their jobs easier. One such tool is the CG Title Entry feature, which enables newsroom personnel to simulate character-generated graphics at the iNEWS Workstation. By offering a graphical view of CG information, iNEWS helps production staff verify the accuracy and quality of the data prior to air.

This chapter contains the following main sections:

- [Overview of CG Title Entry](#)
- [Title Entry Setup and Configuration](#)
- [CG Template Editor](#)
- [Title Entry Security](#)

Overview of CG Title Entry

When writing scripts, journalists are typically required to include information in the script that will appear as a character-generated graphic when the story is aired. These graphics, called supers or CGs, must be entered into the script in such a way that iNEWS can identify the information as data for the character generator. This method of identification uses production cues—also called machine instructions—which are separated from the story text.



The first production cue example appears graphically later in this section.

Here is an example of a production cue for a CG with one line of text for the graphic and one line of information for production staff.

```
*CG live
```

```
USA Hockey Championships
```



Unlike the lines shown here, in iNEWS, production cues that contain machine commands appear as blue text in the Instruction sub-panel on the left side of a split Story panel. For those without machine commands—those that do not begin with an asterisk—the font appears black.

Here is another example of a production cue for a CG with two lines of text for the graphic and two lines of information for iNEWS and production personnel:

```
*CG live
```

```
Ellen Miller
```

```
City Hall
```

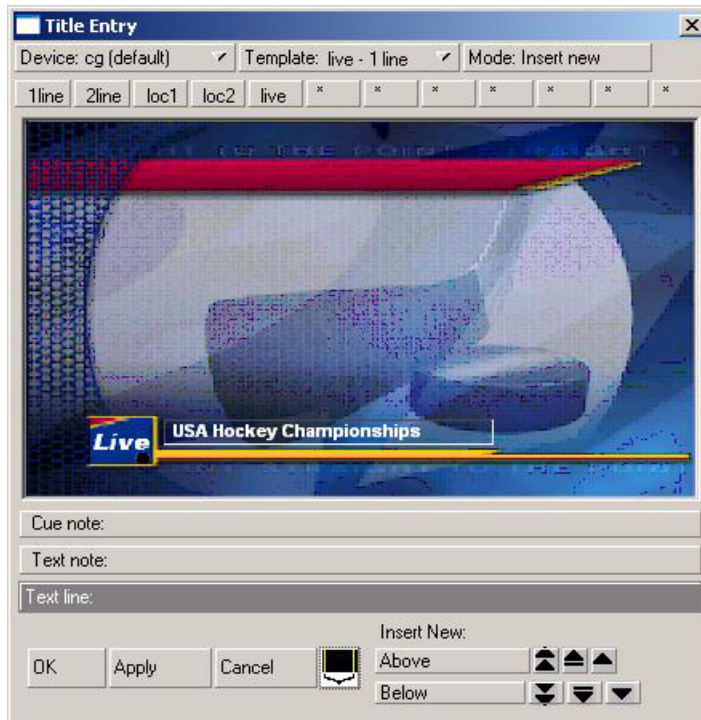
```
Take cg at 10 seconds in...
```

In the example, the user cannot visually determine how information will appear on air when the data is loaded in the character generator template. For instance, will the data fit into the actual text fields on the CG template? The answer to this question cannot be determined from the production cue text. However, the Title Entry feature allows for this possibility by offering a graphic-style dialog box.


The dialog box displays sample templates, complete with backgrounds and text fields, similar to actual CG templates that appear on air. These sample templates—which are configured and modified by system administrators using the CG Template Editor—can be filled in by the users, and information is applied to the production cue within the script.

When configured accordingly, the Title Entry dialog box can provide users with visual verification of whether text they enter will fit in fields of actual character-generated graphics. Producers can use the Title Entry dialog box to preview CG information in scripts in order to check the appearance and accuracy prior to airing the production cue.

The following graphic shows how a single-lined production cue appears in the Title Entry dialog box, using a template background that has a 16 x 9 screen ratio.



The black button at the bottom of the window—known as the Dimensions Toggle button—lets users toggle markers on and off, indicating what part of the graphic would be visible within the standard 4 x 3 screen ratio.

 *The Title Entry feature can only be accessed by users when the cursor is located in the Story panel of the iNEWS Workspace. Access to Title Entry can also be limited to certain users. See “Title Entry Security” on page 241 for more information.*

Title Entry Setup and Configuration

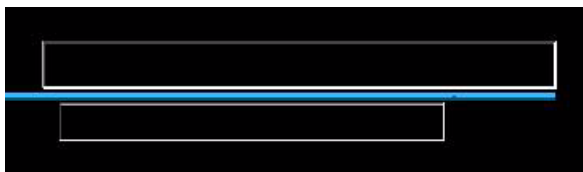
Before you can use the Title Entry feature, templates must be created by the system administrator in iNEWS to simulate the actual graphics used by the character generator. The person responsible for setting up the templates should have a good understanding of how CGs are created on the character generator, because it is important that templates in Title Entry correspond to those used by the character generator.

Understanding CG Templates

Most CGs are comprised of both background graphics and text fields. A background graphic could be a small color bar that spans the bottom third portion of the television screen, a full-screen graphic, a logo in the corner of the screen, and so forth. See [“CG Template Backgrounds” on page 227](#) for more information.


The text fields vary as well. A CG template can contain numerous text fields, with each one having different predefined properties, such as screen location and font style. Typically, users can write text in the fields, but some may contain predefined text that users cannot edit. For instance, a CG can have a field with the predefined word, FILE, which would be aired over file video of previous news coverage.

An example of a standard CG template is a two-line lower-third CG. In the following example, the bar separating the two rectangular boxes makes up the background. The boxes are the text fields, which can be filled in by a user.



 *CG text fields are also known as tab fields, referring to the key used to navigate from one field to another.*

In a CG template, the character generator assigns each text field a number, indicating the order in which fields can be filled in. For instance, a user types a name in field one, then navigates to the second field in the sequence using the Tab key, and types the person’s job title or location.

 *Knowing the numeric field assignments of each template is crucial, because the sequence in the Title Entry dialog box must match the tab field’s sequence on the corresponding character generator’s template.*

CG templates are stored on the character generator at locations assigned or represented by identification numbers known as Template numbers. These numbers can be used by the CG operator to recall templates, which are filled in and aired during a show. However, in iNEWS, aliases are usually assigned to each template to provide users with an easier way of recognizing the template.

For instance, a template containing one field—one line of text—used to air the location of a story may be stored on the character generator as template 502, but users type in the alias `loc1` in the production cue.

The assignment of aliases to Template numbers is stored in the Resource queue located in the System directory (SYSTEM.RESOURCE). These same aliases must be established first and used in naming the templates for Title Entry.



The Title-Entry queue in the System directory (namely, SYSTEM.TITLE-ENTRY) is used to limit access to the CG Template Editor. When the CG Template Editor starts, it gains a lock on SYSTEM.TITLE-ENTRY, which guarantees that only one system administrator can modify the set of templates at any given time. The Title-Entry queue must be created, if it does not already exist, prior to creating templates. In that queue, there is one story titled 000-installation, and that story contains the following text:

```
WARNING - This file is machine read/written - Never edit this text
(unless instructed to do so)
TEMPLATE ROADMAP= TE000001
```

Do not manually edit the data in this queue. For help creating this queue, contact Avid.

When configuring templates or setting up new ones, use the CG Template Editor provided in the software. See [“CG Template Editor” on page 229](#) for more information.

CG Template Backgrounds

Before creating CG templates for Title Entry, the system administrator should capture an image of the actual CG templates on the character generator as bitmaps. These bitmaps are used as background images in the creation process of corresponding templates in iNEWS.



You must create bitmaps in one of two sizes. Standard size is 400 x 300 pixels (for a screen ratio of 4 x 3) with 256 or more colors. The iNEWS CG Title Entry also supports images for a 16 x 9 screen ratio, which requires bitmaps with the larger dimensions of 560 pixels wide by 315 pixels high.

You can store the bitmaps anywhere on the network as long as they are accessible from the iNEWS Workstation used to run the CG Template Editor. When the CG Template Editor program creates Title Entry templates, the chosen bitmaps are copied to the iNEWS database, at which point the bitmaps on the network can be disposed of.

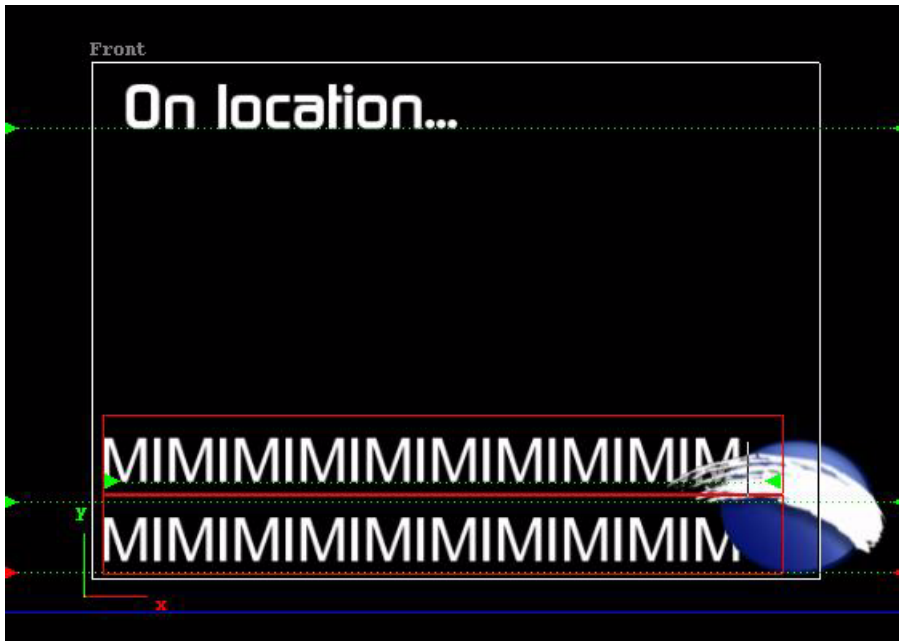
Required Bitmaps

Two slightly different bitmaps of each unique CG template should be captured from the character generator. First, capture the CG template with the text fields full of text.



Do not use spaces.

Each text field in the actual CG template filled out with information from a user should have a solid string of characters in it when the bitmap is made. Fixed-width fonts allow for absolute accuracy; however, proportional fonts can be used. The character limit of a field with fixed-width fonts will be the same no matter which letters of the alphabet are used to fill it up. Proportional—also known as variable-width—fonts contain some letters that are wider than others, so the character limit may vary. For instance, the character limit of a field with a proportional font may be nine letters if “M” is used to fill up the field, but that limit increases to 25 when the letter “I” is used instead. A good mix of wide and narrow letters should be typed in the field to fill it up if a proportional font is used.



The first bitmap provides a graphical demonstration of character width allowed in a text field and the position of that field on the screen. This bitmap will be used to align text fields and fonts on the Title Entry template when created in the CG Template Editor.

Secondly, capture the same CG template with empty text fields as a bitmap.



This bitmap will replace the first one in the final version of the Title Entry template. It will be used as the template background shown in the Title Entry dialog box.

CG Template Editor

System administrators should use the CG Template Editor to create and modify templates referenced by the Title Entry dialog box.



Before starting the CG Template Editor, a character generator and their aliases must be defined in `SYSTEM.RESOURCE`; otherwise, an error stating “No character generators defined” will appear.

To start the CG Template Editor:

1. Log in to iNEWS.
2. Select Tools > CG Templates. For superusers, the Edit Title Entry Template window, which is explained in the next section, will open.



In some cases, a password is required before access to the CG Template Editor is allowed. See “Title Entry Security” on page 241 for more information.

Edit Title Entry Template Window

The Edit Title Entry Template window contains several template options, which can be used to create templates that graphically represent actual CGs aired from the character generator.

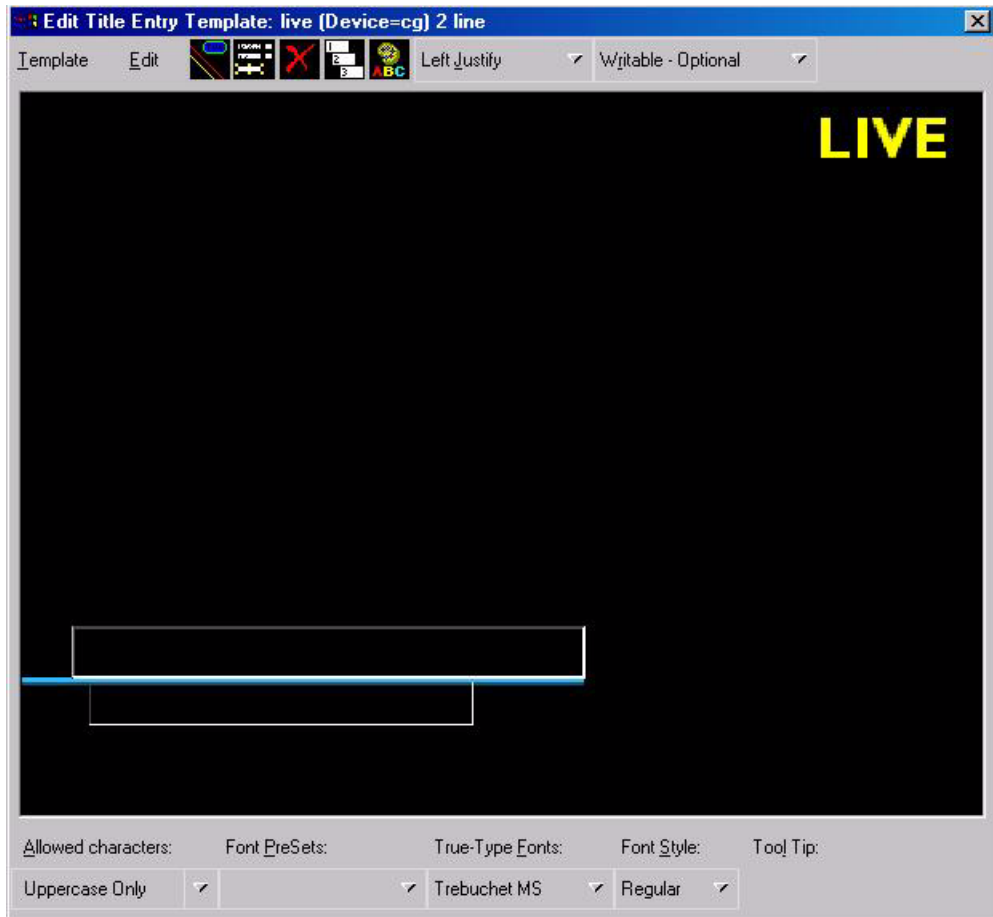
Some options available to the system administrator are:

- Select a bitmap for a CG template background
- Add, modify, or delete text fields in a template
- Reorder sequence of text fields
- Limit editorial access to text fields
- Modify font size, color, style, and so forth
- Provide helpful Tool Tip text, which appears in the Title Entry dialog box when a user positions the mouse over a text field.








Options in the CG Template Editor do not directly affect features of actual CG templates on the character generator, but should be used to correctly reflect to the iNEWS user facsimiles of CG templates on the character generator that are as accurate as possible.

The Edit Title Entry Template window's toolbars—located at the top and bottom of the window—contain various buttons and drop-down menus, used to configure template options.



The following table provides details about the toolbar buttons and menus in Edit Title Entry Template window.

| Menu, List, Field or Button | Options and Explanation |
|---|--|
| Template menu | Options include: New, Open, Save and so forth. This menu is used to open new or existing templates, save modified templates, and to close the window |
| Edit menu | Options include: Background, Add Field, Font PreSets, and so forth. This menu is used for adding or deleting items to a template, such as a background or field, and setting predefined Fonts. |
|  | The Logo button is the same as the Background option in the Edit drop-down menu. It allows the user to select a bitmap graphic as a template background. |

| Menu, List, Field or Button | Options and Explanation (Continued) |
|---|--|
|  | The Text Fields button is the same as the Add Field option in the Edit drop-down menu. It allows the user to add a text field to a Title Entry template, which can then be positioned and sized accordingly. |
|  | The X button is the same as the Delete Field option in the Edit drop-down menu. It allows the user to delete the selected text field from a Title Entry template. |
|  | The 1-2-3 button is the same as the Field Order option in the Edit drop-down menu. It allows the user to rearrange the numerical tab sequence of the text fields in a Title Entry template. |
|  | The Text Color button is the same as the Text Color option in the Edit drop-down menu. It allows the user to change the color of the font used in a text field as displayed in the Title Entry dialog box. |
| Justify list | Options include: Left Justify, Right Justify, and Center Justify. This list is used to set the alignment of text in a field. The drop-down list will continue to display the current setting of the selected field when the list is closed. The default is Left-Justify. |
| Editing list | Options include: Writable, Read Only, Optional and Required. This list allows the user to determine whether a field is Writable or Read Only, and whether it is a required field. If Required is chosen, the user of CG Title Entry will be required to fill out the field with text before saving the production cue. Because the user is prevented from entering text in a read-only field, if Read Only is chosen, the user of the CG Template Editor is required to fill out the field with text before saving the template. Blank, read-only fields are not permitted. The drop-down list will continue to display the current setting of the selected field when the list is closed. The default is Writable-Optional. |
| Allowed Characters list | Options include: All characters OK, Uppercase Only, Lowercase Only, and Define Subset. This list allows the user to limit the type of characters allowed in a text field. For instance, if the text field should only contain numerical sports scores, then the user could use the Define Subset option to limit the allowed characters in that field to the numbers: 0123456789. As shown, do not use a comma to separate the numbers or it too will be allowed in the field. The Define Subset option is case-sensitive. |

| Menu, List, Field or Button | Options and Explanation (Continued) |
|-----------------------------|---|
| Font PreSets list | Options include: 10 user-definable font display settings. This list allows the user to store up to 10 font display settings created by the user as PreSets. These can later be selected when making new templates with fields that contain the same size and style of font, saving the user from having to recreate the font repeatedly. The first font stored is particularly important because it becomes the field properties default for the Template Editor at start up. |
| True-Type Fonts list | Options include: any true-type font installed on the workstation, such as Times New Roman. This list allows the user to select a font that matches or closely resembles the font used on the actual CGs. |
| Font Style menu | Options include: Bold, Italic, Underline, and Strike through. This allows the user to define the style of font used in the selected text field. |
| Tool Tip field | This field allows the user to type in text that will appear as a Tool Tip box in the Title Entry dialog box, when a user positions the mouse over the text field. For instance, if a user should type only a person's name or location in the field, the Tool Tip text for that field could be: <code>Type Person or Location Name Here.</code> |

Creating a New Template

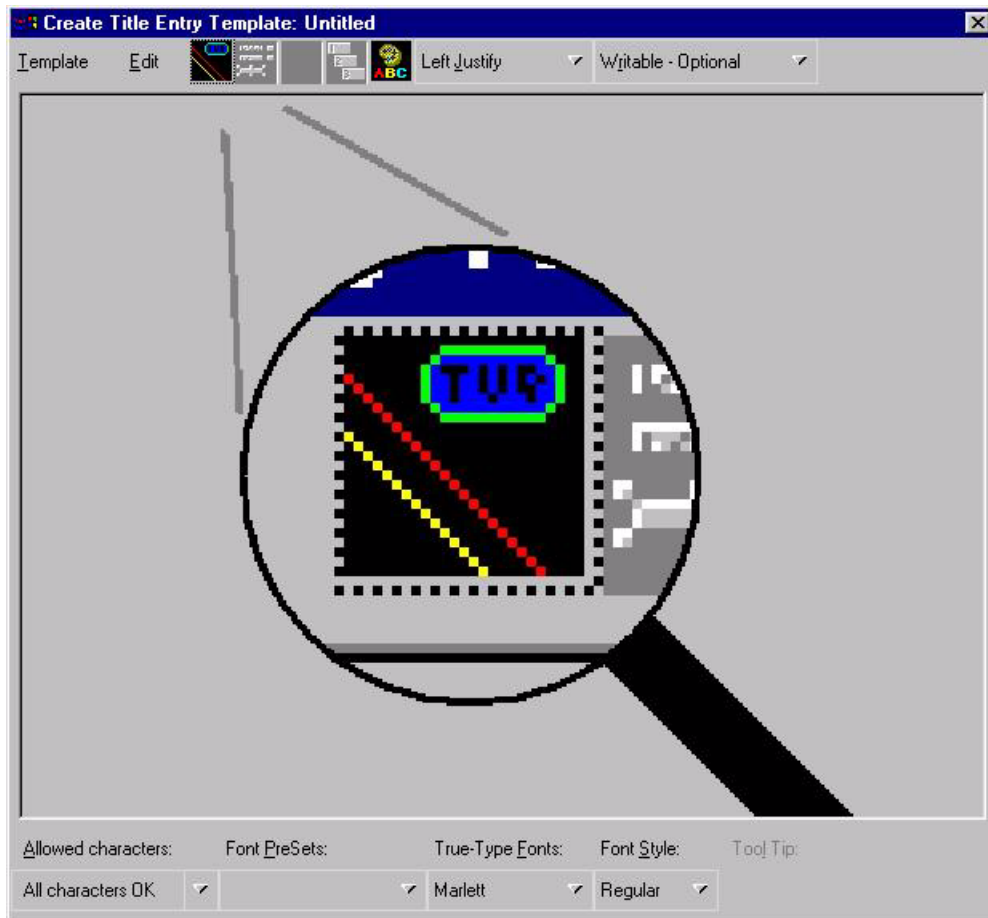
The CG Template Editor is used to recreate a template that will simulate the appearance of information sent to actual templates aired by the character generator. This section assumes those actual CG templates were already captured as bitmaps and stored in a file accessible to the user creating a new template. See [“Required Bitmaps” on page 228](#) for more information.



Since template names must match the alias associated with the actual CG template number used by the character generator, the aliases must be defined before opening the CG Template Editor. For instance, if a user typically types `*CG loc1` as the first line of a production cue for a 1-line location CG, then the Title Entry template created for that CG should also be named `loc1`. Aliases are defined in `SYSTEM.RESOURCE`. The CG Template Editor will not start if there are no available aliases to save templates to.

When the Create Title Entry Template window first opens, it appears with a graphic that highlights the first step in the process for creating new templates.

9 Character Generator Title Entry



To create a new template from the Create/Edit Title Entry Template window:

1. Define a temporary background for the template by doing one of the following:
 - ▶ Click the Logo button.
 - ▶ Click the Edit menu and select Background.



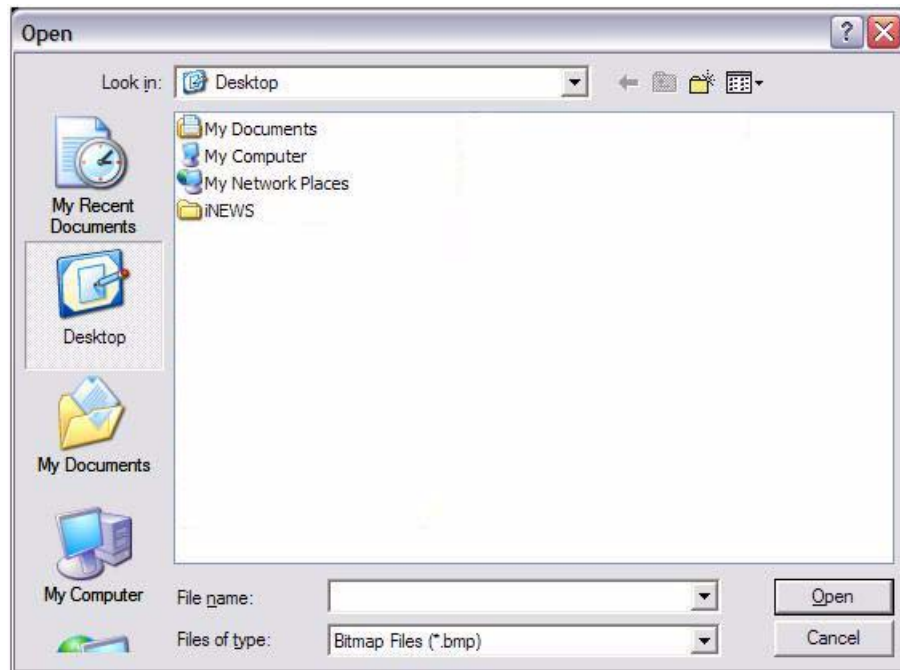
Use the Template menu and select New to create another new template when an existing one already appears in the window.

The Recycle -vs.- New background dialog box appears.



2. Click No.

The standard Windows-based Open dialog box appears.



3. Navigate to the directory where you stored the bitmap background and select a bitmap from the directory where you stored them. This is the bitmap of the actual CG template you want to recreate in the Template Editor. See [“Required Bitmaps” on page 228](#) for more information.



The first bitmap you choose should be the one that contains text fields filled with characters. In upcoming steps, you will use this bitmap to align the text fields and the font in the template, before replacing it with another bitmap containing empty text fields.

4. Add a text field by doing one of the following:
 - ▶ Click the Text Fields button.

9 Character Generator Title Entry

- ▶ Click the Edit menu and select Add Field.



If a template already has a text field and it is selected, then any new field added to the template will follow size and font styles for that original field.

5. Click on the new text field and drag it into position over the text-filled area pictured in the bitmap—one field per line of text in the bitmap. You can also adjust the position of the text field by using any of the following key combinations:
 - Ctrl+Right Arrow move the field to the right
 - Ctrl+Left Arrow moves the field to the left
 - Ctrl+Up Arrow moves the field up
 - Ctrl+Down Arrow moves the field down
6. Use the mouse to resize the text field to reflect the size of the text area in the bitmap.
7. Use the True-Type Font list or the PreSet list to select a font similar to the font used by the character generator, as seen in the bitmap.
8. Type the same characters shown in the bitmap into the new text field. The font may or may not line up exactly over that of the bitmap. To enable Title Entry to accurately reflect the text as it will appear on actual CGs, you must align the field and characters with those shown in the bitmap.
9. Adjust the height and width of the font by using any of the following key combinations:
 - Alt+Right Arrow expands the width of the font
 - Alt+Left Arrow contracts the width of the font
 - Alt+Up Arrow increases height of the font
 - Alt+Down Arrow decreases height of the font

This might take some time and will require experimentation with fonts and font styles, before the characters align with those of the bitmap. The goal is to make certain the text field you create in the template has the same character limit as the area shown in the bitmap when the exact characters are used in both.

10. Use the Backspace key to erase characters from the field once the font and field alignments are complete.
11. Determine the justification of text in each field. Select the field and use the Justify list.
12. Repeat steps 6-11 as needed for each text field in the template.



If text fields on several CG templates use the same font and field size, you do not have to repeat the steps with each new template created. You can duplicate a field (including all its size and font properties) within a template by holding the Control (Ctrl) key down, clicking

on the field, and dragging its copy to a new location within the template. You can also reuse field properties by saving them as a Font PreSet, which can then be selected for use when creating other templates. See “Using Font PreSets” on page 239 for more information.

13. Determine whether a user can write in each field, and whether the user is required to fill in a field by using the Editing drop-down list. If you make a field required, then the user must enter some data into that field when using Title Entry before iNEWS will accept the production cue. This does not apply to the field while in the CG Template Editor. However, if you make a field read-only, it must contain some text so you will be required to put some text in the field while in the CG Template Editor. Users will be prevented from altering that text in the read-only field when using Title Entry.

The Read Only option can be used for setting up templates that standardize certain commonly used CGs. For instance, you have a CG that has two fields: one for any council member’s name and the second that should contain the words, City Council Member. But, users sometimes just write Councilman or worse, they misspell Council. You could make the second field read only and pre-define the words that appear in the field.



If a field is pre-defined on the actual CG template, such as LIVE or FILE, it should appear as part of the background—saved as part of the bitmap—not as a read-only field on the Title Entry CG template.

14. Determine which characters are allowed in each text field. Select the field, then choose from the options in the Allowed Characters drop-down list. If you choose to define your own subset of characters, a dialog box will appear, where you must then type in only those characters you want to allow in the field. For instance, you can limit what a user types into a field for a Sports CG to only characters: **1234OTF**—representing 1st-4th quarter, OT for over-time, and F for final.



*The Allowed Character option is case-sensitive. Also, notice the example in step 15 does not include commas, such as: **1, 2, 3, 4, OT, F**. If commas are included, then commas will also be accepted in the field.*

15. Type in the text for the field’s Tool Tip. The Tool Tip will appear in the Title Entry dialog box when the user positions the mouse over that field. To enter Tool Tip text, click on the Tool Tip field at the bottom-right corner of the Edit Title Entry Template window, then type your text in the field provided. You must press Enter to save the tool tip text.



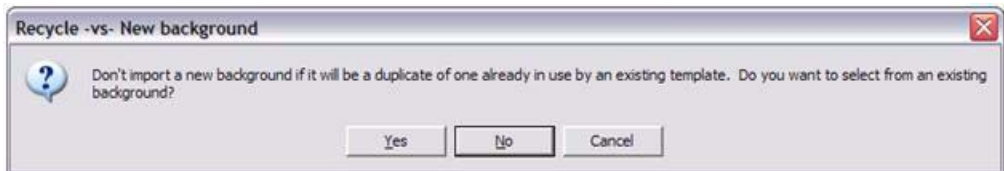
Alt+L can be used to select and enter data into the Tool-Tip field. Also, each Tool Tip will appear with a number in the CG Template Editor window. The number indicates that field’s order in the template. These numbers do not appear as part of actual Tool Tips over fields of the Title Entry dialog box.

9 Character Generator Title Entry

16. Replace the background of the template with the bitmap of the CG Template containing empty text fields. To choose another bitmap, do one the following:

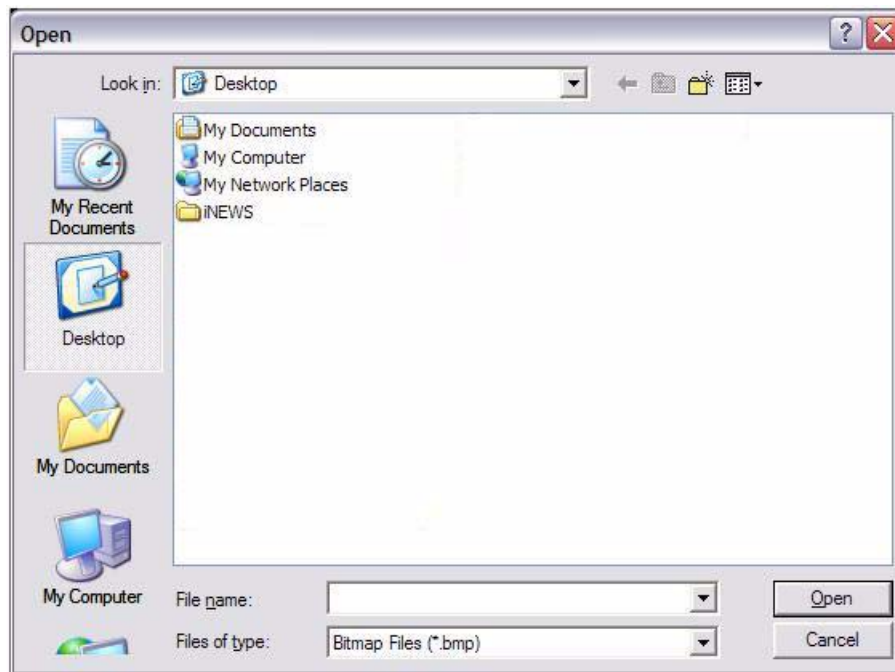
- ▶ Click the Logo button.
- ▶ Select Edit > Background.

The Recycle -vs- New background dialog box appears.



17. Click No.

The standard Windows-based Open dialog box appears.



18. Navigate to the directory where you stored the bitmap background and select the correct bitmap. This is the bitmap of the actual CG template you want to recreate in the Template Editor. See [“Required Bitmaps” on page 228](#) for more information.
19. Save the template by doing one of the following:

- ▶ Select Template > Save.
- ▶ Type Ctrl+S.



The template name must match the pre-defined alias associated with the actual CG template number used by the character generator. Aliases are defined in SYSTEM.RESOURCE.

Using Font PreSets

Font PreSets are a time-saving feature for system administrators using the CG Template Editor to create Title Entry templates.

After a system administrator sets up a text field and saves it as part of a template, he can store settings for that field as a Font PreSet. The Font PreSet can then be selected when the system administrator needs to add a field with those same settings to another template. For instance, the text field of a one-line location CG is saved as a Font PreSet, which is later used to create the first field in a two-line name CG, a two-line live CG, and so forth, because all fields use the same font style and field size.



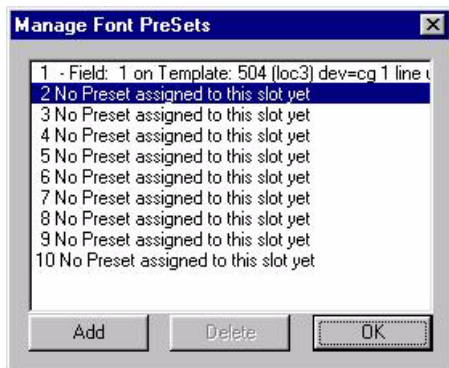
You can have a maximum 10 Font PreSets in iNEWS. The first one stored is particularly important, because it becomes the field properties default for the CG Template Editor at start up. Save the PreSets for major font/field variations. Because you are limited to 10, it is not recommended they be used for minor variations, such as a font color change. You can replace any Font PreSet with a new one if all of the PreSets are taken.

To save a field as a Font PreSet:

1. Open the template containing the field you want to save as a Font PreSet.
2. Click on the specific field to select it.
3. Do one of the following:
 - ▶ Select Edit > Font PreSets.
 - ▶ Press the F7 key.

The Manage Font PreSets dialog box appears with the first available PreSet option highlighted.

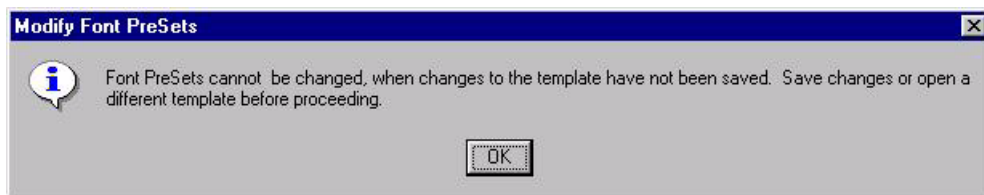
9 Character Generator Title Entry



4. Click Add to save the selected field as a new Font PreSet. The name of the newly added field will appear in place of existing data on the highlighted PreSet option.
5. Click OK to close the dialog box.



You cannot save a field's settings as a Font PreSet until after the field is saved as part of a template. If you attempt to set a Font PreSet using a newly created field from an unsaved template, iNEWS will issue the following advisory.



To delete an existing Font PreSet:

1. Open the Manage Font PreSets dialog box by doing one of the following:
 - ▶ Select Edit > Font PreSets.
 - ▶ Press the F7 key.
2. Select the PreSet you want to delete from the list.
3. Click Delete.
4. Click OK to close the dialog box.

To choose an existing Font PreSet for a selected text field:

1. Click the Font PreSets list at the bottom of the Edit Title Entry Template window.
2. Select the Font PreSet you want to use for the chosen field.

Title Entry Security

Access to CG Template Editor and CG Title Entry is determined separately per user account. A user who can use CG Title Entry will not necessarily have access to CG Template Editor, while another user can be denied the ability to use both features.

Access to CG Template Editor

Access to CG Template Editor is limited to system administrators, or superusers, and those who know the password for dbmanager, if that user account is created by a system administrator. This is the same dbmanager account that is used to modify database traits. See [“The Database Manager Account” on page 97](#) and [“Changing Database Traits” on page 136](#) for more information.

When a non-superuser attempts to launch the CG Template Editor from the Tools menu, the following dialog box appears:



If the correct password is not given, or if the user does not have read access to SYSTEM.TITLE-ENTRY then iNEWS will notify the user and deny access to the Edit Title Entry Template window.



Only one person on the network can use CG Template Editor to edit Title Entry templates at a time.

Access to CG Title Entry

The availability of CG Title Entry is dependent on two things: cursor position in the iNEWS Workspace and permission granted in the user’s account.

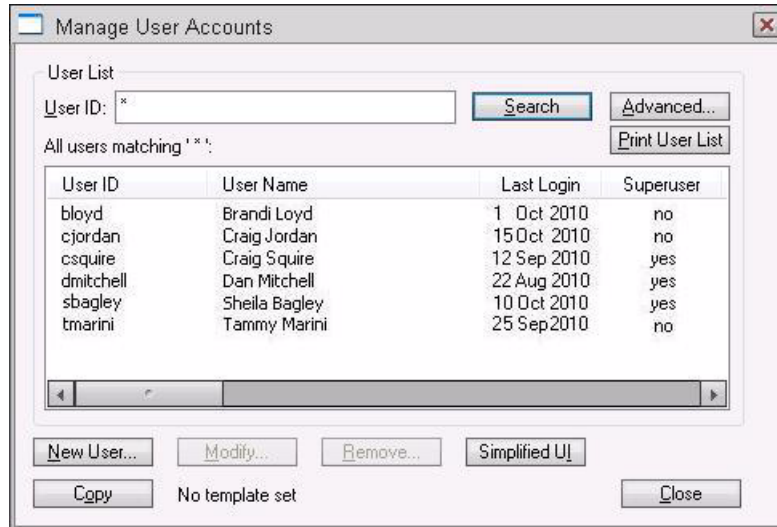
When a user’s cursor is in any panel other than the Story panel, the Titling option in the Tools menu will appear gray, indicating that Title Entry is unavailable. Additionally, a system administrator, or those authorized with the umanager password, can limit access to Title Entry.

There is no individual user setting that enables or prevents access to CG Title Entry. However, those users whose accounts are set as “Simplified” might be prevented from accessing CG Title Entry.

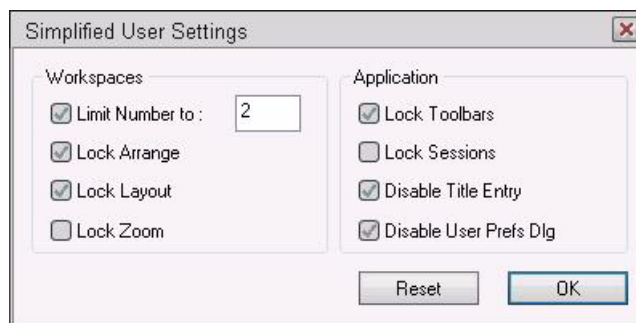
9 Character Generator Title Entry

To prevent access to CG Title Entry for all simplified users:

1. Log in as a system administrator or with your own user account if you know the umanager password.
2. Select Tools > Options > Users. System administrators will see the Manage User Accounts window immediately. Others will be prompted first to give the umanager password before the window opens.



3. Indicate the user by providing the User ID. The Search button is available to help locate the user's account if the ID is unknown.
4. Click the Simplified UI button. The Simplified User Interface dialog box appears.



5. Click on the Disable Title Entry check box to select it.
6. Click OK.

10 System Configuration

Information about your iNEWS system's hardware devices (workstations), connections (wires), and iNEWS Servers is stored in configuration files. This chapter contains a representative sample of each iNEWS configuration file. A comprehensive list of sample configuration files is contained in [“System Files” on page 541](#). Procedures for editing configuration files using the line editor, ed, are provided in [“The Line Editor, ed” on page 655](#).

This chapter contains the following main sections:

- [Overview](#)
- [Configuration File](#)
- [Hosts File](#)
- [System Profile Files](#)
- [Devices](#)
- [Intersystem Messaging](#)

Overview

Configuration of the iNEWS newsroom computer system is controlled by settings stored as text files in the (root) Site directory on the software area of the system's hard disk. Files in the Site directory or subdirectories include:

- /site/config
- /site/system
- /site/wires/<file name>
- /site/dict

Because many system files are located in the Site directory, they are also referred to as site files.



All files in the Site directory, including the system profile and configuration file, can contain only Roman characters.

Another important file is stored in the ETC directory, which is:

- /etc/hosts

Samples of these and other system files are provided in [“System Files” on page 541](#). Changes to system configuration are made by editing these text files. The recommended method is at the console using the line editor, ed, because it works on all servers simultaneously through the console. Unlike familiar graphical-based word processing programs, ed deals with text files on a line-by-line basis.



To learn more about the line editor, refer to one of the books available in bookstores on the Linux operating system with sections devoted to ed.

The most likely reason for you to use ed is to modify system files, such as /site/config or /site/dict/queues. For instance, if you add a wire to your system, you will need to add the wireserver's configuration information to the configuration file in the Site directory. Some procedures for editing certain files are provided in this chapter. For more information about the line editor, type info ed at the console. Additional information, including details on how to use the line editor, ed, is provided in [“The Line Editor, ed” on page 655](#) of this manual.



It is vital that you ensure system files remain identical across all iNEWS Servers. When you modify a site file, make the same changes to each server's copy of the file, or your system will not run properly. Select all servers before you open a file for editing to ensure changes you make are applied to each server's copy of the file.



The Linux graphical X-window login offers additional Linux GUI tools for system management. Tasks such as changing IP addresses and checking hardware are much easier within the X-window GUI. However, if all iNEWS Servers have the GUI installed as part of the Linux installation, it does take additional resources to run and requires additional hardware, such as monitor, keyboard, and mouse. Some system managers might not want to run the GUI full-time so as to reserve memory and other resources for the iNEWS Server processes.

While the impact of running the GUI is probably not terribly significant on modern hardware with speedy processors and plenty of RAM, Avid does not test the software or system performance with the GUI running. This editing alternative is not required or supported by Avid.

Before editing any system file, Avid recommends you make a backup copy of the file.

Making a Backup File

When you want to make changes to a system file, begin by making a backup copy of the file, and then edit the backup file. That way, if you make a mistake during the editing process, your original file version is preserved.

To make a backup copy of a file:

- ▶ Use the copy command at the console in the following format:

```
cp <file pathname> <new pathname>
```

For instance, to copy the configuration file in the Site directory, type:

```
NRCS-A$ cp /site/config /site/config.backup
```

Viewing System Files

When viewing a system file, use the more command at the console.

The format is: `more <file pathname>`

The more command allows you to view one page at a time, which is especially useful for really long files. The cat command may also be used, particularly for smaller files.

To use the more command to view your copy of the configuration file:

- ▶ Type: `NRCS-A$ more /site/config.test`

Information similar to the following example appears:

10 System 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
```



Only part of the configuration file is shown here. An entire example is available in “System Files” on page 541.

To use the cat command to view the system profile, a text file stored in /site/system:

▶ Type: NRCS-A\$ `cat /site/system`

Information similar to the following example appears:

```
NRCS-A$ cat /site/system
id=INWS      net=abc
lowwater=100000  highwater=105000  purgelimt=5
load=5       pausetimeout=0:05
;
; iNEWS 2008-07-31 16:01:20 - Added wordlength parameter
wordlength=6
;
; defaults - if parameter not present these values are assumed
;           unless overridden on the command line.
;
; auto_upgrade=yes      lowwater=100000      remotetimeout=0:00
; clockmax=12           maxhits=500
; excluded_video=none   min_passwd_length=5
; highwater=105000     msgserver=silent      security=or
; lastlogin=yes        pausetimeout=0:30      timechar=:
; load=0               purgelimt=0          timer=verbose
; localtimeout=0:00    readrate=180
```

Licensing iNEWS Components

Each time the iNEWS newsroom computer system is configured, your licensing information is checked. This information determines the number of devices you are authorized to connect to the system. An error message appears if the configuration file defines more devices than are licensed in any of the following categories:

- iNEWS workstations (sessions)
- COM
- Web Access
- Web Client
- Web Services API
- Wires

10 System Configuration

- Instinct
- Community

To display your system's licensing limits, at the console, type:

▶ NRCS-A\$ **status license -OR- status 1** (lowercase L)

A message similar to the following will appear on your screen:

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

System is AB. Master is A.

Disk status is OK. The database is OPEN.

Site Key.....: 009999

CPUs.....: 3

Workstation addresses: 3000

Workstation resources: 1000

COM resources.....: 5

Web Access resources.: 2

Web Client resources.: 10

Web API resources....: 5

Wire Server resources: 8

Instinct resources...: 10

Community Sessions...: allowed.

The Workstation addresses category indicates how many IP and/or MAC addresses can be specified in the SYSTEM.CLIENT.WINDOWS story. The iNEWS Server will only accept a connection from a workstation that is identified in the appropriate story; otherwise, an error is given stating that the workstation is “not authorized to connect.” If Workstation addresses shows a "site" license, the SYSTEM.CLIENT.WINDOWS queue should be deleted, unless IP-specific restriction is still desired.

The resources category defines the total number of simultaneous login sessions. The Wire Server resources line indicates the number of wire server resources allowed for the Avid Data Receiver.



When the system is configured, if the number of IP or MAC addresses present in the SYSTEM.CLIENT.WINDOWS story exceeds the number of addresses, a diagnostic is produced. The system will be configured. This differs from exceeding other license limits.

To change license allowances:

- ▶ Contact an Avid sales representative.

Configuration File

The configuration file (/site/config) is a system road map. It lists all devices, servers, and resources configured to run on your system and how they are connected. If a device is not in the site configuration file, the system will not know about it and cannot use it. Standard devices and resources you may configure in this file include monitors, iNEWS Workstations, and wire services.

Each server for your system has a copy of this file that it reads when it starts up and when you execute the configure command. However, it is only the configuration file on the master computer that is active and used when the system is started up.



Whenever you make changes to a site file, such as the configuration file, be sure to select all servers in your system at the console. Unlike database stories, site files are not automatically mirrored from one computer's disk to another. See “Selecting One or More Servers” on page 45 for more information. Also, see “The Line Editor, ed” on page 655 for information on using the line editor, ed, to make changes to site files.

An example of a configuration file is located in “System Files” on page 541.

The site configuration file is divided into two major sections: the host section and the device section (or body). The host section contains information about various configurations your system can run, and devices used in each of those configurations.

The format for each host section is:

```
host <system configuration> <computer>
      <reslist>
      <servers>
```

| Parameter | Description |
|----------------------|---|
| system configuration | Specifies in what configuration this section will be used. For example, a 2-server system normally runs AB. But if one server fails, the configuration would be A or B. The config file will contain entries for all possible system configurations. |
| computer | Specifies the responsibilities of this particular computer in the configuration. So if the system configuration is AB (for a dual-server system), and the computer is A, this section will define the A computer's responsibilities. See “Selecting One or More Servers” on page 45 for more information. |

| Parameter | Description (Continued) |
|-----------|--|
| reslist | Refers to resources configured on that particular iNEWS Server in that system configuration. |
| servers | Refers to various utility programs called servers that are configured to run on that particular iNEWS Server in that system configuration. This term should not be confused with the computers, also called servers, which run the iNEWS application software. See “Servers” on page 315 for more information. |

The top host section details the device, resource, and utility program numbers that run on the A server in a dual AB configuration. The second host section details ones assigned to server B.

Information in the third and fourth host section is used by the system if one of the servers fails. In the sample site configuration file, the “host a a” section contains a list of all the devices, including ones normally assigned to server B. If server B experiences a failure and is shut down, the system can be reconfigured to run all devices, resources, and servers (utility programs) on A. The “host b b” section contains a list of all the devices, including ones that normally run on server A, in case that server experiences a failure and is shut down.

When you run the configure command, the master computer (usually server A) looks at the current system configuration and then assigns devices listed for each iNEWS Server in that system configuration to each iNEWS Server.

In the sample /site/config file, the odd numbered devices are assigned to server A and even numbered devices are assigned to server B in a dual AB configuration.

Any item number listed in the host section must have a corresponding line in the device section or body of the configuration file, and vice versa. For instance, if you are adding an iNEWS Workstation resource to the body of the file, you must also add it to one or more host sections so the system knows which server would be responsible for it under various conditions.

Here are the possible device configuration lines:

```

server      301      monitor  301      -      ; monitor
server      401      rxnet    -        -      ; RXNET server
server      501      txnet    501      -      ; TXNET server
server      601      action   601      -      ; action server
server      801      ftsindex 801      -      ; FTS index server
server      901      ftsseek  901      -      ; FTS seek server
server      902      ftsseek  901      -      ; FTS seek server (notice mailbox #)
server      1101     keyword  1101     -      ; keyword server
server      1201     mailserver 1201     -      ; mail server
server      1401     seek     1401     -      ; seek server
aiws        1601     -        gnews    -      ; Interplay session
api         1701     -        gnews    -      ; Web Services API session
cinws       1801     -        gnews    -      ; Community session
com         1901     -        gnews    -      ; COM session
iiws        2001     -        gnews    -      ; iNEWS-licensed Instinct session
inws        2101     -        gnews    -      ; iNEWS workstation session
resource    2301     console  -        -      ; connect session
webclient   2401     -        gnews    -      ; Web Client session
websession  2501     -        -        -      ; Web Access session
wireserver  2601     news     AP       -      ; Wire server session
wireserver  2602     news     BC       join   ; Wire server session with join enabled

```

A colon can be used to specify value ranges in device configuration lines, as shown in the following example:

```

inws        2101:2200 -        gnews    -      ; 100 iNEWS workstation sessions

```

10 System Configuration

Also, individual devices can be restricted to specific IP or MAC addresses, as shown in the following examples:

| | | | | | |
|--------|------|-------------------|--------------|---|-------------------|
| server | 1301 | rxnet | 172.24.96.44 | – | ; INWS-B |
| inws | 2101 | 00:19:B9:0E:7D:E1 | gnews | – | ; control room PC |
| inws | 2102 | 17.24.96.247 | gnews | – | ; director laptop |

Editing the Configuration File

Whenever you add, remove, or modify devices on your system, you must make corresponding changes to the configuration file—also referred to as the `/site/config` file. Changing this file requires the use of the line editor. See [“The Line Editor, ed” on page 655](#) for more information.

Before editing any system file, Avid recommends you make a backup copy of the file.

To edit the configuration file:

1. Select all servers.
2. Type: `ed /site/config`

After you press Enter, the editor responds by displaying a numerical value indicating the file size expressed as the number of characters, including spaces and returns.



This procedure, which modifies the `/site/config` file uses `ed`, the line editor. If you do not know how to use `ed` to modify lines in the file, please see [“The Line Editor, ed” on page 655](#).

3. What information is edited in the file depends on the device. Editing procedures for specific devices are provided later in this guide. For more information, see [“Adding Devices” on page 265](#).
4. Type `w` to write (save) your changes to disk.



Do not use an uppercase `W` in this step. Uppercase `W` will append the file you're editing to the existing file. The resulting file may be unreadable and will lead to problems with running your iNEWS system.

5. Type `q` to quit the line editor.

Testing the Site Configuration File After Alteration

Whenever you make changes to `/site/config`, *always* run a test on the changes to ensure there are no problems with the new configuration. By doing so, the test will warn of problems or if license limits are exceeded. Some configuration problems will prevent system configuration and startup.

To run the site configuration test:

- ▶ Use the configure console command.

The format is: `configure <config file> <system> <computer>`

System refers to the set of servers that make up your iNEWS system, while computer is the server whose configuration you have changed.

The following example shows the command for testing the configuration of server A in an AB system:

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

When the prompt returns, the configuration file has been checked. If the system detects any errors, it displays bad configuration messages.

Incorporating Configuration Changes

After testing the configuration, you can incorporate any changes that you made.

To put the new configuration into effect:

1. Stop any devices affected by the new configuration.
2. Take the system offline by typing: `NRCS-A$ offline`
3. Configure the system, using the following format: `configure`
4. Bring the system back online by typing: `NRCS-A$ online`
5. Wait for messages from the system being configured, and then restart the newly added devices or any devices affected by the new configuration.

To list contents of the site configuration file at the console:

- ▶ Type: `cat /site/config`

The configuration file that appears on your screen is similar to the sample file provided in [“System Files” on page 541](#).

Semicolons precede comments and blank lines separate sections.

The sample file to which we refer throughout this explanation might not match your system’s file exactly, but it contains examples of the different kinds of entries you might find in your file.

10 System Configuration



On lines of the /site/config files where device numbers are listed, a range of numbers may be specified—two numbers separated only by a colon (:). This is useful at sites with many resources, servers, and sessions. For instance, the resource line below shows device numbers listed individually, then the line is repeated, showing how it might appear as a range.

```
reslist 100 100 102 103 104 105 106 107 108 109
```

```
reslist 100:109
```

Similarly, you can specify a device range for definitions, such as:

```
inws 201:210 - gnews - ; iNEWS Workstation sessions
```

This is equivalent to 10 lines, each with a different device number, with all other information on the line applying to each device definition. If an IP address is specified, a device range cannot be specified.

Hosts File

The hosts file (/etc/hosts) is a road map to other computers on the network that your iNEWS Servers need to know about or communicate with. It lists IP addresses of other computers and the computers' names, along with any alternate names or aliases by which the other computers are known. Workstations need not be listed in the hosts file.

If a computer is not listed in the hosts file, your iNEWS Servers will require DNS to find them. Your iNEWS system may not be configured to use DNS. Also, if DNS fails, your iNEWS Servers will be unable to use it to find external computers. By putting the external computer in /etc/hosts, you make the connection more reliable..

An example of a hosts file is located in [“System Files” on page 541](#).

System Profile Files

For your system's servers to work together, they need access to basic system information, such as how many computers are in the system and how the computers are connected. Each server must also have access to system parameters, such as the default read rate and script margins. This information is kept in system profile files whose names reflect the information they contain. For instance, information about wires is kept in /site/wires.

Sample system profile files are in [“System Files” on page 541](#). The most important of these profile files is the system profile(/site/system), which is discussed in the sections that follow. When you start your system, each server reads its copy of the system profile and incorporates the material into its operation.

An example of the system profile is provided in [“Viewing System Files” on page 245](#).

The system profile contains several parameters. Each parameter begins with an identifying keyword, followed by an equal sign (=) and the parameter’s value. When an iNEWS Server reads its system profile, it finds each parameter’s value by searching for the keyword that represents that parameter. If the server searched the system profile in the example on [page 245](#), it would find that NRCS was the parameter associated with the keyword, id.

Most, but not all, parameters have default values the system uses if the parameter is not present in the system profile. Consequently, a system profile usually includes only parameters you want set differently from their default values, and those that have no default values and must be set in the system profile. Your system profile may not contain the same parameters as the example.

Changing the System Profile

System profile parameters are configured in the system when servers are connected. After the system profile in `/site/system` is modified, the system should be shut down and restarted to get the servers to read the modified system profile. The following sample procedure lets you change the `localtimeout` parameter in your system profile.

To change your system profile at the console:

1. Select all servers.
2. Type: `ed /site/system`



This procedure, which modifies the `/site/system` file uses `ed`, the line editor. If you do not know how to use `ed` to modify lines in the file, please see [“The Line Editor, `ed`” on page 655](#).

A message similar to the following appears:

```
editing /site/system
213
```

3. Find the line that contains the `localtimeout` parameter, such as:


```
localtimeout=45:00 remotetimeout=30:00
```
4. Change the 45:00 value for that parameter to 15:00:


```
localtimeout=15:00 remotetimeout=30:00
```
5. Type `w` to write (save) your changes to disk.



Do not use an uppercase `W` in this step. Uppercase `W` will append the file you're editing to the existing file. The resulting file may be unreadable and will lead to problems with running your iNEWS system.

10 System Configuration

6. Type **q** to quit the line editor.



When you modify your system profile, separate parameters from each other with spaces, tab spaces, or carriage returns.

7. Shut down and restart your system.

When you start the system, each server reads its system profile and incorporates parameters in that file in its operation.



You must reboot the servers and connect the system to get the servers to read the system profile file.

Listing Parameter Settings

The status all command lists values for all parameters defined in the profile, except for the low and high watermarks and the purge limit.

To find out which parameters the system is using:

- ▶ At the console, type: **status all**

This command displays the system profile settings the system has incorporated in its operation. Information similar to the following appears:

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

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

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

```
System was last configured at 2009-08-10 13:53:49
```

```
No dbtraits group changes since boot.
```

```
No ctraits changes since boot.
```

```
auto_upgrade=yes    maxhits=500
clockmax=12         min_passwd_length=5
excludedvideo=none  pausetimeout=0:05  security=or
lastlogin=yes       readrate=180        timechar=:
load=5              remotetimeout=0:00  wordlength=6
localtimeout=0:00
```

Parameters not explicitly defined in the system profile will appear in the list with their default values.

To list current low and high watermarks and the purge limit:

► Type: `cat /site/system`

See “[Viewing System Files](#)” on page 245 for more information.

System Profile Parameters

Each system profile description below includes information about the values that parameter can have and whether or not the parameter has a default value. If a parameter does not have a default value, you must give it a value in the system profile.



The connect command can override anything in the profile by adding `<parameter>=<value>` on the connect command line.

| Parameter | Description |
|--|--|
| <code>auto_upgrade=[yes no]</code> | Determines whether a user running an outdated version of the iNEWS Workstation (client) software is allowed to upgrade it automatically. If set to no, it means users of outdated software are not asked if they want to upgrade. The default is yes. |
| <code>clockmax=[12 24]</code> | Determines how backtimes/cumetimes (cumulative) are displayed—12 or 24-hour format. While the iNEWS Workstation always displays these times in 24-hour format, this setting is used by various utility (server) programs whenever a date/time is formatted, such as when an action server processes a mailto command and a story is sent as an email message. The default value is 12. |
| <code>excludedvideo=[director none]</code> | Determines the handling of director video when received in Story Exchange Protocol (SEP) format. If set to none, this text is converted into bold italic text. If set to director, the text is converted into closed caption (CC) text. The default is none. |
| <code>highwater=<# of block units> (105000)</code> | Establishes the upper limit to which dobserver attempts to rebuild the free list. Set this parameter far enough above the low watermark so the system is not in danger of slipping beneath that mark. The number you enter represents actual blocks of database space, where a block is one kilobyte. For instance, a high watermark of 125000 (recommended for most systems) represents 125000 blocks. The default value is 105000. |

10 System Configuration

| Parameter (Continued) | Description (Continued) |
|---|---|
| <code>id=<system name></code> | <p>Gives the system a name. The system uses this name in some of its messages and in the prompts. It also appears in the iNEWS Workstation representing the database. The system name defined here must match that used in the <code>/etc/hosts</code> file (minus the “-a” or “-b”), must be in uppercase, and can be up to eight characters long. There is no default value, so you must set it in the system profile.</p> |
| <code>lastlogin=[yes no]</code> | <p>Lets you suppress on a user’s workstation display of last time user was logged in. Setting no accomplishes this; the default is yes.</p> |
| <code>load=<number of connections> (0)</code> | <p>Specifies the maximum numerical difference the system tries to maintain between network connections from Windows-based clients on different servers in your system. This is called load balancing, and it is intended to keep one server from handling a much higher number of connections than any other server.</p> <p>For instance, if you set this parameter to 5, the system distributes connection requests so the difference in the number of connections is no higher than 5. If you had 2 active connections on server A, and 7 on B, the next request to connect to B would be shifted to A. Connection requests for A would be allowed, until the number of connections on A was 5 more than on B.</p> <p>The default value for this parameter is 0, which means load balancing does not occur.</p> |
| <code>localtimeout=<min:sec> (00:00)</code> | <p>As a security precaution, your system automatically logs out workstations if they are idle longer than the time set in this parameter.</p> <p>If a story is open at an idle workstation, the system saves the story before logging out the workstation. Setting <code>localtimeout=00:00</code> prevents the system from logging out workstations. This is the default setting.</p> <p>The <code>localtimeout</code> parameter is set in minutes and seconds. For instance, to have the system log out any workstation idle for more than two hours, set this parameter to <code>120:00</code>. The maximum value is 540 minutes.</p> |

| Parameter (Continued) | Description (Continued) |
|---|--|
| lowwater=<# of block units> (100000) | <p>Establishes minimum disk space that the system tries to keep available for immediate use. Use it with the highwater and purgelimit parameters to control how the system recycles space in the database. Set this parameter in units of actual database blocks, where a block is one kilobyte (1KB). For instance, a low watermark of 100000 (recommended for most systems) equals 100000 blocks of database space.</p> <p>If the number of blocks in the free list falls below the low watermark, the system runs dbserver to reclaim the oldest stories from the Dead queue, recycling the space onto the free list. This continues until the free list is restored to the high watermark. If you do not include this parameter, the system uses the default value of 100000 blocks.</p> |
| master=<computer name> | <p>Designates one of your system's servers as the master computer, which controls all database activity and performs the majority of housekeeping, such as running dbpurge every hour and invoking dbserver when low on space.</p> <p>Generally, this parameter is left out of the system profile, causing the system to designate as the master computer the server whose name is alphabetically first (usually server A). You can specify a server as the master computer using the connect console command. The format is:</p> <pre data-bbox="682 991 1093 1012">connect <computer> master=<name></pre> |
| maxhits=<number of hits> (500) | <p>Defines maximum number of hits, or matches, that a background search—including Fast Text Search (FTS)—can find. For instance, setting this parameter to 50 limits the total number of hits in a single search to 50. The maximum number you can specify is 32765. If you do not assign a value, the system uses a default value of 500.</p> |
| min_passwd_length=<# of characters> (5) | <p>Defines minimum password length for your users. For instance, setting min_passwd_length=6 prevents users from creating passwords shorter than six characters. The value may range between 1 and 12 characters. This does not apply to passwords you assign with the utraits command. The system uses 5 as the default value.</p> |

10 System Configuration

| Parameter (Continued) | Description (Continued) |
|---|---|
| <code>msgserver=[silent verbose]</code> | Used only for debugging. To find out whether or not a mailbox is working, set this parameter to verbose. This causes the console to display a message whenever activity occurs in a queue with a mailbox assigned to it. Change this parameter while the system is running using the <code>msgdebug</code> command. The default setting is silent. This prevents messages regarding mailbox activity from being displayed. |
| <code>name=<computer name></code> | Each server must have a unique name (either A, B, C, or D) to distinguish it from the other servers in the system. Typically, assign these names during the startup process using the <code>connect</code> console command— <code>connect a</code> for server A, and so forth—so it need not appear in the system profile. There is no default value assigned. |
| <code>net=<computer names></code> | This allows you to specify all servers in the network. For instance, in a system using three servers named A, B, and C, this parameter would be set to <code>net=abc</code> in the system profile. |
| <code>pausetimeout=<min:sec> (00:30)</code> | Sets a default value for the PAUSE command, which is used in some keyboard definitions. Users can override this default value. If not set, the system assumes a value of 30 seconds. |
| <code>purgelimit=<hours> (0)</code> | <p>If <code>dbserver</code> reclaims all space available in Dead queue without restoring the free list to the low watermark, it begins to purge old stories by making a series of passes through the database. On each pass, <code>dbserver</code> temporarily decreases each queue's purge interval by one hour and removes any stories older than the new purge interval. It continues doing this until it has rebuilt the free list to the high watermark or reaches the purge limit.</p> <p>The purge limit sets the maximum number of passes <code>dbserver</code> can make through each queue. The total number of hours <code>dbserver</code> can purge from a queue is equal to the queue's purge interval minus your system's purge limit.</p> <p>Use the purge limit to prevent <code>dbserver</code> from purging everything from important queues in its attempt to build up the free list. For instance, if you set the purge limit to two hours, queues with a 3-hour purge interval retain at least one hour's worth of stories, even in a low-on-space situation.</p> <p>You can set the purge limit between 0 and 24. If not set, your system uses a default value of zero hours, which prevents <code>dbserver</code> from purging any queue beyond its purge interval.</p> |

| Parameter (Continued) | Description (Continued) |
|---------------------------------|---|
| readrate=<words/minute> (180) | Sets the system's default read rate. When you add a user to the system, the Add New User dialog box will default to a read-rate of zero, which will be replaced by the /site/system read rate when needed. After a user has been added, you can change the user's read rate using the Modify User Account dialog box. If not set, the system uses a default read rate of 180 words per minute. |
| remotetimeout=<min:sec> (00:00) | This time-out value applies to all connect sessions, including sessions that connect a workstation at your station to another service. Disable the automatic logout of connect sessions by setting this parameter to 00:00. The maximum value is 540:00. The default that the system uses is 00:00 if this parameter is not included in the system profile. |
| security=[and or] | Indicates how your system determines group access of a particular user with a particular workstation (or other device). If set to "and", both user and workstation must be members of the same group for the user to gain access to directories or queues assigned to that group. If set to "or", the user can access any database items assigned to groups containing either the user or workstation. The default value is "or." |
| single=<computer name> | Tells the system it is running on only one server and names that server. If your system consists of a single server, include this parameter. Generally, in systems with only one server, the name is A, and this parameter is set to single=a. |
| timechar=<character> (:) | Defines the character the system uses to separate hours, minutes, and seconds in time displays. For instance, using a colon as the time character displays the time as hh:mm:ss. A colon is the default. |
| timer=[silent verbose] | Your system contains a timer program that is always running. If this parameter is set to verbose, the server sends a time display to the console every 15 minutes. |
| | A Sat Apr 3 14:45:00 2004 iNEWS |
| | The default setting is verbose, so do not include when you want timer messages to appear. To disable time displays, include this parameter as timer=silent. |



All time values in the system profile must be set in minutes and seconds in the format <mm>:<ss>. Set a value for both minutes and seconds. For instance, to specify a time of two minutes, type: 2:00. To specify a time of 25 seconds, type: 0:25.

| Parameter (Continued) | Description (Continued) |
|----------------------------|---|
| wordlength=<numeric value> | This is the number of characters that are counted as one word. When counting words for script timing, the system divides the character count by the wordlength to determine the number of “words” in the story. The value is required and must be a whole number. |

Devices

The iNEWS newsroom computer system supports the following devices:

| Device | Description |
|------------------|---|
| Instinct | A connection used by some Instinct and NewsCutter clients to communicate with the iNEWS Server. |
| Web Services API | A connection used by Web-based services to communicate with the iNEWS Server. |
| Web Access | A connection that allows direct, read-only browsing of the iNEWS database through the Web browser. |
| Community | A connection used by clients from other systems in your iNEWS Community to connect with your iNEWS Server. |
| COM | A (connection) device used by some Avid applications, such as MOS Gateway, to communicate with the iNEWS Server. |
| Workstation | The PC device on which a user can log in to the iNEWS newsroom computer system known as an iNEWS Workstation. |
| Web Client | A connection through which a Web Client user can log in to the iNEWS newsroom computer system. |
| Network Resource | A device that allows a user to connect to another computer system from a workstation. |
| Server | A utility program that performs tasks on stories in a queue, based on defined instructions. Supported server types include: action, distribution, parallel wire, monitor, and keyword. Other servers facilitate searches, mail, and printing, or are designed for Rx/Tx network links—receiving or transferring stories between computer systems. |

See “Servers” on page 315 for more information.



The server listed here is not the same as the computers used as iNEWS Servers, typically given names with the station’s call letters and an A, B, or C. For instance, NRCS-A.

| Device | Description |
|------------|---|
| WireServer | A device that allows an Avid Data Receiver instance to connect to the iNEWS newsroom computer system. |

Viewing Information about Devices

The `list c` console command prints information to the console screen about configuration of a device or devices. The format for this command is:

```
list c [<device type, device number, or program name>]
```

The device type or device number is optional. If you do not enter them, you get a list of the configuration information for all devices in your system.

To view information about devices connected to your system:

- ▶ Use forms of the `list c` console command.

For instance, when you type `list c` at the console, information similar to the following appears:

| DEV | DEVICE_TYPE | COMPUTER | NOTIFY | OPTIONS | DEVNAME |
|------|-------------|----------|--------|---------|---------|
| S201 | mailserver | A | 201 | | |
| S211 | keyword | A | 211 | | |
| S231 | seek | A | 231 | | |
| S241 | monitor | A | 241 | | |
| ... | | | | | |
| R391 | console | A | | | |
| B401 | websession | A | | | |
| G501 | gnews | A | | | |
| w601 | news | A | | | AP |
| g701 | gnews | A | | | |
| A711 | gnews | A | | | |
| C801 | gnews | A | | | |
| a811 | gnews | A | | | |

10 System Configuration

...

If you follow list `c` with the name of a program, iNEWS lists every device on your system that uses that program. For instance, to find out how many devices use the action server utility program, type: **list c action**

Information similar to the following appears:

| DEV | DEVICE_TYPE | COMPUTER | NOTIFY | OPTIONS | DEVNAME |
|------|-------------|----------|--------|---------|---------|
| S251 | action | A | 251 | | |
| S252 | action | A | 252 | | |
| S253 | action | A | 253 | | |
| S254 | action | A | 254 | | |

Another example is **list c monitor**, in which case configuration information for your system's monitor servers are displayed, similar to the following:

| DEV | DEVICE_TYPE | COMPUTER | NOTIFY | OPTIONS | DEVNAME |
|------|-------------|----------|--------|---------|---------|
| S241 | monitor | A | 241 | | |
| S242 | monitor | B | 242 | | |
| S243 | monitor | A | 243 | | |
| S244 | monitor | B | 244 | | |

If you want to know the configuration information for a specific device, use the device number, such as typing: **list c 344**

Information similar to the following appears:

| DEV | DEVICE_TYPE | COMPUTER | NOTIFY | OPTIONS | DEVNAME |
|------|-------------|----------|--------|---------|---------|
| S344 | rxnet | A | | | |

List C Message Columns

There are six columns of information displayed in list `c` messages, as explained in this table:

| Column Name | Description |
|-------------|---|
| DEV | Lists the device number. The number is preceded by a letter identifying the type of device. These alphabetic identifiers are as follows: A - Avid Instinct and Newscutter a - Web Services API B - Web Access C - Community c - COM device G - iNEWS Workstation g - Web Client R - Network resource S - Server (utility program) w - Wire server |
| DEVICE_TYPE | The program that runs the device. For instance, workstations have gnews listed in this column. |
| COMPUTER | Identifies the iNEWS Server on which the device is running. |
| NOTIFY | Identifies the number of the mailbox the device uses to receive notifications. |
| OPTIONS | Lists any modifiers to the device, such as IP or MAC address. |
| DEVNAME | Lists the device's name, if it has one, which can be used for group membership and placed in the DEVNAME field of stories. |

Adding Devices

When you add a device, such as workstations or wires, to your iNEWS system, you need to put information about it in the appropriate configuration file(s). This section gives you the complete procedure for adding workstations.



The iNEWS system receives wire services through an Avid Data Receiver wireserver resource. For more information about wires, see “Wires” on page 291. For more on how to set up wire ingest using the Avid Data Receiver, see “Adding a Wire – Avid Data Receiver” on page 292.

Also, for information about adding connect services, see the iNEWS Operations and Troubleshooting Manual.

To add a new workstation to your system:

1. Choose a device number for the workstation—determine the next available number in the range you have set aside for these devices.
2. Connect the workstation to the network.
3. Add workstation to the configuration file stored on each server in your system.

Add the workstation's device number to a reslist line in the host definitions.

To configure a workstation, use the format:

```
inws <device #> <address> gnews <device name>
```

| Parameter | Description |
|-------------|--|
| device # | Workstation's device number. |
| address | <p>If you use a hyphen (-), this resource is available to any licensed PC on your system. If you place an IP address here (example: 172.24.96.247), this resource can be used only by the PC with that address.</p> <p>Additionally, this can be the MAC address assigned to the PC's network card (example: 0019b90e7de1).</p> |
| device name | <p>A device name up to eight characters used for group security. If you assign a device name to a workstation, you can grant security permissions to the workstation by adding that name to the appropriate group story.</p> <p>When someone edits and saves a story at this workstation, its device name is placed in the devname field (if one exists) in the Story Form panel. If you do not want to give the workstation a device name, place a hyphen (-) in this position.</p> |

4. (Optional) Use the configure command to test your configuration changes.

The syntax is: `configure /site/config <system> <computer>`

For instance, type: `configure /site/config ab a`

When prompt returns, the configuration file has been checked. If the system detects any errors, it displays appropriate bad configuration messages.

5. Reconfigure the system.
 - a. Select the master computer, which is typically server A.
 - b. Enter superuser mode, using the current password.
 - c. Take the system offline by typing: `NRCS-A# offline`
 - d. Reconfigure the system by typing: `NRCS-A# configure`

- e. When the prompt reappears, bring the system back online by typing:
NRCS-A# **online**
 - f. Press Ctrl+D to leave superuser mode.
The pound sign (#) at the end of the console's server prompt will change to a dollar sign (\$).
6. (Optional) Back up site files with the sitedump command any time you add a device.

Intersystem Messaging

Intersystem Messaging is a feature letting users exchange messages across separate iNEWS newsroom computer systems, or an other third-party system with a compatible interface.

On iNEWS systems, intersystem messages can be sent from iNEWS Workstation sessions, and from the console send utility. For intersystem messaging to work, a system must have an agent that functions as described in the following section. For iNEWS, this agent is integrated into the iNEWS Server software.

To receive intersystem messages, a system must have a TCPMUX service running. TCPMUX is defined by RFC 1078, "TCP Port Service Multiplexer (TCPMUX)." Additionally, the system must have an intersystem message service configured.



RFC (Request For Comments) documentation is provided at the following Web sites:
<http://sunsite.auc.dk/RFC> or <http://www.rfc-editor.org/>

Sending Intersystem Messages

An intersystem send is attempted whenever a message send request has a recipient name which includes an at symbol (@). It is assumed that this represents a name in `<user name>@<system name>` format. This is the same format used for sending mail to a user on a foreign system, such as the Internet. The `<system name>` parameter can be an IP address in standard notation, such as 172.161.131.2. When using Community messaging, the `@<system>` information can be excluded.

Your iNEWS Server will use its `/etc/hosts` file or DNS, if configured, to find the external system.

A TCP connection to port 1 of the system is attempted. Port 1 is the "well known port" (as defined in RFC 1700, "Assigned Numbers") assigned to the TCPMUX service.

After the connection is established, the string `inter_system_message<cr-lf>` is sent. The receiving system sends `+<explanation><cr-lf>` to indicate a positive acknowledgement. This conforms to RFC 1078.

10 System Configuration



The service name—in this case, `inter_system_message`—is never case-sensitive and `<explanation>` is any text that helps to explain the reason for the response.

The sending system can then send the following string:

```
SEND<sp><user name><sp><sender's name><sp><message text><cr-lf>
```

In this string, the user name and sender's name do not contain any spaces, carriage returns, or line feeds. The names are as they are used within their respective systems. The sender's name should be suitable to use as the user name in an intersystem message reply. The message text can contain spaces but not carriage returns or line feeds. It is optional (so you can check the logged-in status of the user). The iNEWS system will truncate this string at the first line feed or at 72 characters.

After sending the intersystem message, the sending system should read a single line, which is the receiving system's response. On receipt of the response, the sending system should close its connection. See [“Receiving Intersystem Messages” on page 268](#) for more information.

The sending system should be prepared to handle all of these error conditions:

- Time out on establishing connection to the receiving system's TCPMUX port
- The receiving system actively refusing the connection on the TCPMUX port
- Connection closed by the receiving system at any time
- A negative response `-<explanation><cr-lf>` to the `inter_system_message<cr-lf>` request. This is a negative TCPMUX response, and `<explanation>` is any text that helps to explain the reason for the response.

Receiving Intersystem Messages

To receive intersystem messages, a system must respond to connections on the TCPMUX port. On Linux systems, this is done by having a TCPMUX service defined in `/etc/services` and `/etc/inetd.conf`.

To hook an intersystem message service into the TCPMUX service on a Linux system, an entry must be included in the `/etc/inetd.conf`, such as:

```
tcpmux/+inter_system_message stream tcp nowait <agent> <parameters>
```

The `inter_system_message` string is the identifier used by the sending system to select this service. This string is not case sensitive.

The plus character (+) preceding `inter_system_message` tells the `inetd` daemon to handle the initial connection and negotiation. In this case, when the `inetd` daemon determines that it has an intersystem message agent, it will perform the positive acknowledgement (the `+. . .<cr-lf>` response) and then invoke the agent program.

The agent program must be prepared to respond to the “SEND” command, as described above.

The iNEWS agent program is `/exc/ismessage`. The suggested parameter for the iNEWS intersystem message agent is `ism`. This parameter can be anything and is used to identify the program in messages printed to the system’s console. If an additional parameter is supplied and it is a non-zero decimal string, the `/exc/ismessage` program is put into a verbose mode. When in verbose mode, the `/exc/ismessage` program will print its responses onto the system console. This can be used to track frequency and identity of intersystem messages.



The actual message text is not printed onto the console.

For iNEWS, the `/etc/xinetd.d/ismessage` file contains:

```
# default: on
# description: The ism server serves iNEWS InterSystem Message requests.
```

```
service ismessage
{
id            =iNEWS-ismessage
disable       =no
flags         =REUSE
socket_type   =stream
wait          =no
user          =root
server        =/exc/ismessage
log_on_failure+ =USERID
}
```

Responses from the receiving agent program must conform to the following syntax:

```
<3-digit response code><sp><explanation><cr-lf>
```

The 3-digit response code is modeled on FTP response codes. (See Section 4.2 of RFC 959, “File Transfer Protocol”.)



However, only single line responses are expected. The explanation is any text excluding a carriage return – linefeed (cr-lf), which makes the response better understood.

The receiving agent program will generate one of the following responses, with the iNEWS receiving agent including the following explanations:

| Response | Description |
|--|---|
| 201 <user name> is logged in<cr-lf> | Message stored for the specified user (if there is message text) and the user is currently logged in. User is notified of message arrival. |
| 202 <user name> is not logged in<cr-lf> | Message stored for the specified user (if there is message text) and the user is not currently logged in. |
| 421 System not online<cr-lf> | System not connected, not configured, or not online. Message is discarded. |
| 430 No such user: <user name><cr-lf> | Username unknown on receiving system. Message is discarded. |
| 450 Message save failed for: <user name><cr-lf> | Failed to properly store the message for the specified user. Message is discarded. |
| 500 Syntax error, command unrecognized<cr-lf> | The first “word” on the line was not “send.” The check for the word “send” is case insensitive. Message is discarded. |
| 501 Syntax error, insufficient parameters<cr-lf> | The “send” line has fewer than three space-delimited tokens. Minimally “send”, <user name>, and <sender’s name> are required, <message text> is optional. Message is discarded. |

The iNEWS receiving agent will print diagnostics to the system’s console when abnormal conditions are encountered. The diagnostics are:

| Diagnostics | Description |
|---|--|
| ism: getpeername failed (<errno> <errno string> | The sender’s IP address could not be determined. |
| ism: fgets error (<errno> <errno string> | The read failed for the send command. |
| ism: gethostbyaddr failed (<errno> <errno string> | The sender’s IP address could not be converted into a host name. |

Errno is the Linux system error number returned on system function calls, and errno string is an explanation of that error code.

The iNEWS receiving agent will accept intersystem messages directed to the user's computer. Messages addressed to "computer" will be printed on the system's console. The word "computer" can be localized using the message dictionary token M_COMPUTER. A 201 computer is logged in response will always be returned for messages directed to "computer."

iNEWS Workstation Session Behavior

There is virtually no difference between sending local messages and sending intersystem messages. If the recipient's name contains an at symbol (@), name validation is not performed and an intersystem send is attempted.

The only difference on receiving messages is that the complete sender's information is always returned. If it is a local message, a simple user name is provided. If it is an intersystem message, the sender's name will be formatted as <sender's name>@<system name>.

11 Printers

Managing your local printer settings is covered in this chapter.

This chapter contains the following main sections:

- [Local Printing](#)
 - [Local Printing Dialog Box](#)
- [Creating and Using Print Styles](#)
 - [Local Print Style Options](#)
 - [Banner Format Options](#)
 - [Example Style Story](#)

Local Printing

Local printing is any print request sent to a printer directly accessible to the iNEWS Workstation. The iNEWS system includes an enhanced Local printing feature, also called Windows printing. These enhancements include a dialog box that offers users more options, such as selecting a predefined style when printing locally.

When local printing a batch of stories, stories that have a BLANK body are not printed. You can also have iNEWS skip a story when printing by putting an asterisk (*) in the story's PAGE-NUMBER field, located in the Story Form or Queue panel.

Local printing does require some setup preparation or alterations. For instance, there are print style options that must be defined in SYSTEM.STYLES. Both print style options and the dialog box are explained here, starting with the dialog box.

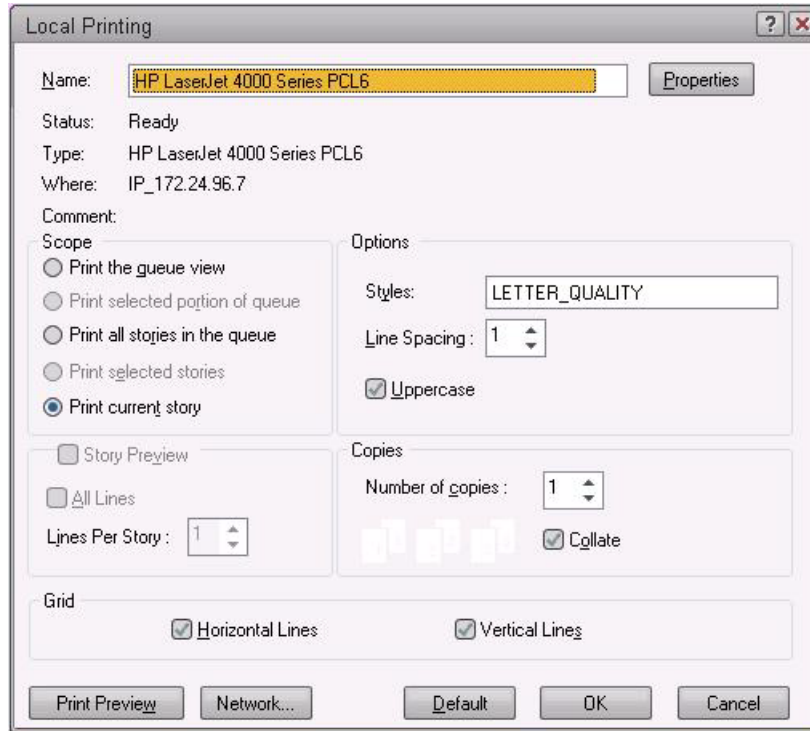
Local printing is not available when the cursor is in the Directory panel, unless it is on a queue that is currently displayed in the Queue panel.



Local printing is not recommended for stories larger than 10 kilobytes or 5 minutes, according to iNEWS timing. The iNEWS system will print these files, but because the local printing is not handled in the background, it might cause the workstation to appear “frozen” or “locked up” while the system processes the print request.

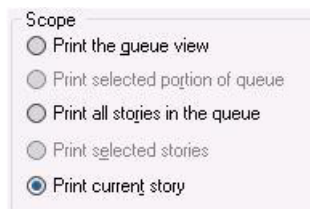
Local Printing Dialog Box

The dialog box is divided into five sections: Scope, Options, Story Preview, Copies, and Grid. There are also several buttons, which include: Print Preview, Network, and Default. Each of these sections and buttons are described in more detail in this part of the chapter.



Scope

The Scope section allows users to select one of four radio buttons to indicate what they want to print.



The options these radio buttons offer are explained as follows:



The “Print the queue view” option determines whether the Story Preview checkbox is available. See “Story Preview” on page 336 for more information.

- **Print the queue view** - This option is available in most cases. A user’s cursor does not have to be in the Queue panel of the iNEWS Workspace for this option to be used. Even if the cursor is located in the Story panel, the user can still print the queue view by selecting this option.
- **Print selected portion of queue** - This option is available when a portion of the Queue panel is selected. Only that highlighted portion of the queue view—which must be contiguous—will be printed when this option is chosen. This option is not available when noncontiguous lines in the queue are selected.
- **Print all stories in the queue** - This option is available from the Directory panel or Queue panel if the queue has the printable (+p) database trait. This option will send the text of all stories in the queue to the printer.
- **Print selected stories** - This option is only available if a row or rows of stories are selected in the Queue panel. The option will send text of stories highlighted in the queue to the printer.
- **Print current story** - This option is available in all three panels of the iNEWS Workspace. It will send the story that currently appears in the Story panel to the printer.

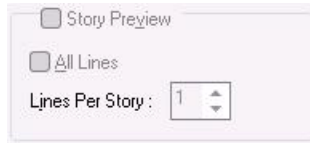
Not all options are available all the time. Also, which option in the Scope section is selected by default depends on which panel in the iNEWS Workspace is activated—that is, in which panel the cursor is located.

Default Option Selection:

- From the Directory panel, if the queue has the printable database trait, it defaults to “Print all stories.” If the queue does not have the printable database trait, it will default to “Print current story.”
- From the Queue panel with one or more stories selected by the row selector button, it will default to “Print selected stories.”
- From the Queue panel with no stories selected, it will default to “Print current story.”
- From the Story Panel, it will default to “Print current story.”

Story Preview

The Story Preview section allows a user to print a preview or sample of each script as part of the queue image. Consequently, this section is only enabled if the “Print the queue view” option is chosen in the Scope section of the dialog box. Otherwise, it will appear gray and be unavailable.



The Story Preview section, once enabled, has three options a user can define, depending on how the queue view should be printed.

- Story Preview - When this check box is selected, iNEWS will print lines of each story, along with the queue view, as defined by the user in the Story Preview section.
- All lines - When this check box is selected, iNEWS will print all of lines of each story, along with the queue image. When this option is selected, the remaining option called Lines Per Story appears gray and is unavailable.
- Lines Per Story - When enabled, iNEWS will print the specified number of lines of each story, along with the queue view. Lines of the story will be printed below the row that corresponds to it in the queue view. This option is disabled (appears gray) when the All lines option is checked.

For instance, if a user wants to see the first few lines of each story in addition to a show's lineup, the user selects the "Print the queue view" radio button in the Scope section of the dialog box, selects the Story Preview check box, sets the Lines Per Story setting to 3, and clicks OK to print it. The iNEWS system will print the first row of the lineup (or fields of the Story Form), followed by three lines of that story, then the second row and three lines of the second story, and so forth.

Rows as shown in the Queue panel may not match exactly rows of the queue image printed with Story Preview selected. That is because iNEWS' Story Preview printing feature is configurable. System administrators can define what information appears in the columns of a Queue panel and in fields of the Story Form panel. These two displays may or may not match exactly. This is also the case for Story Preview printing. For instance, a system administrator can designate which fields from the Story Form are not printed. The default is to print all fields in the Story Form, as specified within each story.

Options

The Options section of the dialog box contains three choices for users: Styles, Line Spacing, and Uppercase.

11 Printers



- Styles - A user can choose from a drop-down list of predefined styles.



Various print style options can be defined by the system administrator and are stored in the first story in the queues within SYSTEM.STYLES. See “Local Print Style Options” on page 280 for more information.

- Line Spacing - A user can configure how much space appears between lines of text. A setting of one would result in single-spaced text, two would be double-spaced, and so forth. This option will override any line spacing predefined in the chosen print style.



The maximum number for line spacing allowed by default in the dialog box is 10. A print style can exceed this maximum and that setting will appear in the Line Spacing box when that style is chosen. However, if a user chooses to manually override the style setting by changing it in the Local Printing dialog box, then the default maximum of 10 will again take effect.

- Uppercase - When this check box is selected, iNEWS will print all text in uppercase (all capitalized letters). Whether this check box appears selected by default when the dialog box first opens can be set by the system administrator using a print style option. See “Local Print Style Options” on page 280 for more information.

Copies

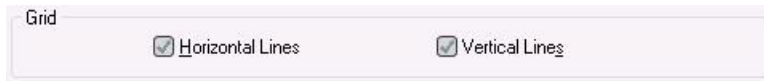
The Copies section allows the user to determine the number of copies printed and whether multiple pages are collated.



The Collate check box is not available for selection unless the number of copies is set to a number more than 1.

Grid

The Grid section of the dialog box contains two choices: Horizontal and Vertical.



- Horizontal - When this check box is selected, iNEWS will include horizontal grid lines when printing the queue view image.
- Vertical - When this check box is selected, iNEWS will include vertical grid lines when printing the queue view image.

The default behavior of these check boxes is based on a user's preferences.

Print Preview and Network buttons

Print Preview - This button allows users to preview queues or stories on their screens prior to printing.



The Print Preview button is disabled when Story Preview is selected.

- Network - This button offers the user more flexibility in selecting which printer is used. The dialog box has two locations from which a user can select a printer. The first is a drop-down list at the top of the dialog box that displays a list of printers loaded on, or locally connected to, the computer the user is on. The second is the Network button, which allows the user to select any printer available through the network.
- Default - This button returns settings of the Print dialog box back to default preferences of the user.

Creating and Using Print Styles

For your printer profiles to be useful, you must have styles designed to take advantage of them. Print styles can be used to define printed output. Each of your system's styles is defined in a separate story in your database, called a style story. Each style story creates a particular kind of printed output by selecting forms and fonts necessary to produce the output.

Each style story is held in its own queue in SYSTEM.STYLES. Queues in this directory have names that are three-digit numbers, such as 001. This lets you refer to each style by a number. For instance, a user can choose the print style defined in the style story in queue 001 by selecting that queue.

Each three-digit queue name in the example can be followed by a hyphen and a descriptive name, such as SINGLE-SPACE. This optional name is used by the Local Printing dialog box.

To create a new style story and queue:

1. Create the new queue in SYSTEM.STYLES. See [“Creating a New Queue” on page 104](#) for more information.
2. Choose the next available three-digit number for the queue name. For instance, if 004 is the last queue name used, create a style 005. Give the new queue a numerical name, such as 005-Memo. The number is followed by a hyphen and the descriptive word, Memo, which will appear as a style option in the Local Printing dialog box.



The largest number that you can use as a queue name is 255 and the lowest is 0. (Style 0 is the default style your system uses when someone enters a print command without specifying a style.) You can have as many as 250 separate style stories.

3. After the queue is created, open it by double-clicking on it in the Directory panel.
4. Create a new story to hold the style you want to create. See [“Creating a New Story” on page 111](#) for more information.
5. Create the new style by writing the various print style and banner options in the Story panel. This is explained in more detail in the next sections.
6. Save the story.

Local Print Style Options

Local print style options are stored in the first story in queues within SYSTEM.STYLES.

In the story, options are specified with an option name—also known as a token—followed by an equal sign (=) and then the parameter value. All options for local printing begin with the letters WIN. Each option must be on its own line. A semicolon (;) will cause the remainder of that line to be ignored, and can be used to make notes in reference to the print option. The format looks like this:

```
<option name>=<parameter value>;<notes>
```

A sample story containing local print styles is provided later in this document. See [“Example Style Story” on page 288](#) for more information.

The local print options are not case-sensitive, but do have default settings provided in the following table:

| Option | Parameter Value | Description |
|-----------------------|-----------------|---|
| WinVTEmulation | YES NO | Specifies whether blank lines will be added to the script side to line up the production cue and the marker (grommet) if the production cue spans more than one line. The default is NO and will keep local printing functioning without inserting blank lines. |
| WinRundownFontFace | | Specifies the font to be used when printing the queue view. Underscores cannot be used to represent a space in a font name. If a font name is not specified, the currently displayed font on screen will be used. |
| WinRundownFontSize | | Specifies a numeric value for the size of the font in points when printing the queue view. If a font face and font size is not specified, the currently displayed font size on screen will be used. If a font face is specified but the font size is not, the font size will default to 12. |
| WinRundownLeftMargin | <left margin> | Specifies a numeric value for the left margin when printing the queue view. Measured in inches. Default size for the left margin will depend on the width of the queue view. |
| WinRundownRightMargin | <right margin> | Specifies a numeric value for the right margin when printing the queue view. Measured in inches. Default size for the right margin will depend on the width of the queue view. |

11 Printers



| Option (Continued) | Parameter Value | Description (Continued) |
|---------------------------|--------------------|---|
| WinRundownTopMargin | <top margin> | Specifies a numeric value for the top margin when printing the queue view. Measured in inches. Default size is 1. |
| WinRundownBottomMargin | <bottom margin> | Specifies a numeric value for the bottom margin when printing the queue view. Measured in inches. Default size is 1. |
| WinRundownColumnHighlight | YES NO | Specifies whether to include shading of column headers when printing a rundown. Default is YES. |
| WinRundownBorder | YES NO | Specifies whether to print a border around the rundown. Default is YES. |
| WinBannerBlanks | <# of blank lines> | Specifies the number of blank lines printed in between the banner and the start of the story form. Default is 0 (zero). |
| WinPrintRundownForm | <form> | Defines what form is used when printing the queue view. If a form is not specified, the currently displayed form on screen will be used. |
| WinPrintHide | YES NO | Specifies whether to prevent certain text in the story body from printing. Default is NO. See WinPrintHideStart and WinPrintHideStop on how to define what text does not get printed. |
| WinPrintHideStart | <character string> | Specifies a character or character string that will be used to mark the beginning of a portion of text in the story body that is not to be printed. Numbers and letters cannot be used. Default is <<. |
| WinPrintHideStop | <character string> | Specifies a character or character string that will be used to mark the end of a portion of text in the story body that is not to be printed. Numbers and letters cannot be used. Default is >>. |
| WinWYSIWYG | YES NO | Specifies whether to use WYSIWYG printing in the printout. If YES, then the font style/size for both the story form and story body will be overridden by the font style/size currently being used on screen for the form and body. Default is NO. |

| Option (Continued) | Parameter Value | Description (Continued) |
|---------------------|----------------------------|--|
| WinSegmentPageBreak | YES NO | Specifies whether to force a page break in the queue view printout at “Break” lines. Default is NO. |
| WinSpacing | <# of line spacing> | WinSpacing supports the use of decimals. For example, WinSpacing=1.5 can be used to define line spacing of 1 1/2 lines. Default is 1. |
| WinColorPrint | YES NO | Specifies whether to use color printing in the story form on printouts. Default is YES. |
| WinPageFooter | <queue name>.<footer name> | Specifies the use of a footer. The <footer name> must match the name of a story in a queue, as defined by the <queue name>, located in SYSTEM.STYLES. Default is NONE. |
| WinPageHeader | <queue name>.<header name> | Specifies use of a header. The <header name> must match the name of a story in a queue, as defined by the <queue name>, located in SYSTEM.STYLES. Default is NONE. |
| WinHeaderFontFace | | Specifies font used in the header and/or footer on the printout. Underscore cannot be used to represent a space. Does not recognize bold, italic, or underline. Default is MS Sans Serif. |
| WinHeaderFontSize | | Specifies a numeric value for the font size, in points, used in the header and/or footer on the printout. Default is 10. |
| WinBanner | YES NO | Specifies whether to include the banner in printout. Default is yes. |
| WinLeftBanner | <format> | Defines banner format options (such as user name, page number, and queue name for printout) on left side of banner. See “Banner Format Options” on page 286 for more information. |
| WinRightBanner | <format> | Defines banner format options (such as user name, page number, and queue name for printout) on right side of banner. See “Banner Format Options” on page 286 for more information. |

11 Printers

| Option (Continued) | Parameter Value | Description (Continued) |
|---------------------------|------------------------|---|
| WinBannerFontFace | | Specifies font used in the banner on the printout. Underscore cannot be used to represent a space. Does not recognize bold, italic, or underline. Default is MS Sans Serif. |
| WinBannerFontSize | | Specifies a numeric value for the font size, in points, used in the banner on the printout. Default is 10. |
| WinForm | YES NO ALL | Specifies whether a form is included in the printout of each story. The YES option specifies printing the form on first page of each story; NO specifies none; ALL specifies printing the form on every page of each story. |
| WinPrintForm | <form> | Defines which form is used for printing. Default is form specified within each story. |
| WinFormFontFace | <form> | Specifies font used in the form on the printout. Underscore cannot be used to represent a space. Does not recognize bold, italic, or underlined text. Default is MS Sans Serif. |
| WinFormFontSize | | Specifies a numeric value for the font size, in points, used in the form on the printout. Default is 10 |
| WinSeparator | YES NO | Specifies whether a horizontal rule (line) is printed after the Story Form row and before the line of text from the story. Default is NO. |
| WinBlanks | <# of blank lines> | Specifies a numeric value for blank lines printed following each story when printing the Story Preview. Default is 2. |
| WinSpacing | <# of line spacing> | Specifies a numeric value for the line spacing of printed stories and queues. Default is 1, which results in a single-spaced printout. |
| WinScriptShift | YES NO | Specifies whether iNEWS selects the Uppercase checkbox in Local Printing dialog box by default, and prints all text in uppercase. Default is YES. |

| Option (Continued) | Parameter Value | Description (Continued) |
|---------------------------|------------------------|---|
| WinBodyFontFace | | Specifies font used in body of the printout. Underscore cannot be used to represent a space. System will print bold, italic, or underlined text, as indicated in the original text of story. This option also applies to printed production cues. Default is MS Sans Serif. |
| WinBodyFontSize | | Specifies a numeric value for the font size, in points, used in the body of the printout. Also applies to printed production cues. Default is 10. |
| WinOrientation | <orientation> | Specifies an override of page orientation in the Printer Properties dialog. Allowable options are portrait or landscape. Default is as specified by the Printer Properties dialog. |
| WinScriptSplit | YES NO | When enabled, stories are printed in a dual-column layout regardless of whether the story has any production cues if the user chooses to print all stories, selected stories, or the current story. Default is NO. |
| WinScriptLeftWidth | <left width> | Specifies a numeric value for the width—measured in inches—of the left side of the story that contains production cues. Default is 3. |
| WinScriptRightWidth | <right width> | Specifies a numeric value for the width—measured in inches—of the right side of the story that contains story text. Default is 4. |
| WinLeftMargin | <left margin> | Specifies a numeric value for the left print margin—measured in inches. Default is .50. |
| WinRightMargin | <right margin> | Specifies a numeric value for the right print margin—measured in inches. Default is .50. |
| WinTopMargin | <top margin> | Specifies a numeric value for the top print margin—measured in inches. Default is 1.0. |
| WinBottomMargin | <bottom margin> | Specifies a numeric value for the bottom print margin—measured in inches. Default is 1.0. |

| Option (Continued) | Parameter Value | Description (Continued) |
|--|---|--|
|  | <i>The sum of the defaults: WinLeftMargin + WinScriptLeftWidth + WinScriptRightWidth + WinRightMargin = .50 + 3 + 4 + .50 = 8.0. This width does not exceed a standard page of 8.5.</i> | |
| WinSequentialNumbering | YES NO | <p>When enabled, if the user chooses to print the Queue panel or a portion of the Queue panel, the pages of the queue view will be numbered 1, 2, 3, and so forth. Any information in the page-number field of each story will be printed as it exists in the system. When a user chooses to print all stories, the story pages will be numbered 1, 2, 3, and so forth with the numbers continuing across stories. When a user chooses a selection of stories, the story pages will be numbered 1, 2, 3, and so forth with the numbers restarting at each story. If a user prints a current story, the story pages are also numbered 1, 2, 3, and so forth.</p> <p>When the Copies option is set to >1, the page numbering will be repeated on each copy of a story regardless of whether the Collate option is selected. For instance, if the pages of story #2 are numbered 4, 5, and 6, each copy of that story will have pages 4, 5, and 6.</p> |
|  | <i>The page number will only be printed if f-page-number is listed as one of the attributes to include in the WinLeftBanner or WinRightBanner options. This is true regardless of whether the user decides to enable the WinSequentialNumbering option.</i> | |
| WinForceCollate | YES NO | <p>When enabled, multiple-copy print jobs are collated regardless of whether the specified printer accepts the Collate option if selected. Default is NO.</p> |
| WinCCShading | <comma-delimited RGB value> | Defines shading of closed captioning. |
| WinPIShading | <comma-delimited RGB value> | Defines shading of presenter instructions. |
| WinMIShading | <comma-delimited RGB value> | Defines shading of machine instructions. |

Banner Format Options

In Local printing, iNEWS will include a banner at the top of printed text if the style selected by the user includes the WinBanner=yes option.

The banner comes in two parts—a left side and a right side—each with its own user-configurable option. The system administrator can set what information appears in either side of the banner by using the local printing style options `WinLeftBanner` and `WinRightBanner`.



Literal text can be used with `WinLeftBanner` and `WinRightBanner` parameters. For instance, if you want the banner to print the word, “User” followed by a user’s name, define the parameter this way: `WinLeftBanner=USER-&u-name`

See “Example Style Story” on page 288 for other literal text examples.

These options recognize parameters only in certain formats, as shown in the following table:

| Parameter | Description |
|---------------------------------|---|
| <code>&u-name</code> | user name |
| <code>&q-name</code> | queue name |
| <code>&date</code> | date (according to Windows regional setting) |
| <code>&time</code> | time (according to Windows regional setting) |
| <code>&f-title</code> | slug |
| <code>&f-page-number</code> | page number |
| <code>&f-presenter</code> | presenter name (according to the presenter field) |

To define a banner in a local print style:

1. Navigate to `SYSTEM.STYLES` in the Directory panel.
2. Double-click on the queue you want to modify, opening it in the Queue panel.
3. Double-click on the first story in the queue, opening that story in the Story panel.
4. Position your cursor in the Story panel.
5. Press Enter to ensure your cursor is at the start of a new line.
6. Type `WinBanner=yes` on the new line. If this line already appears somewhere in the style, you can skip this step.
7. Type `WinLeftBanner=` followed by a `<format parameter>`. Separate multiple format parameters with single spaces. Format parameters are shown in the preceding table.

For instance, if the left side of the banner should include the user’s name and date, the local print style option would look something like this:

```
WinLeftBanner=&u-name &date
```



The left banner print style option should be defined even if it will be blank; otherwise, the system uses the default setting. To define the left banner as blank, type: WinLeftBanner= leaving the remainder of the line after the equal sign blank. The left banner default setting is: WinLeftBanner=&u-name &date &time

8. Press Enter to make certain your cursor is at the start of a new line.
9. Type **WinRightBanner=** followed by a *<format parameter>*. Separate multiple format parameters with single spaces. Format parameters are shown in the preceding table.

For instance, if the right side of the banner should include the story slug and queue name, the local print style option would look something like this:

WinRightBanner=&f-title &q-name



The right banner print style option should be defined even if it will be blank; otherwise, the system uses the default setting. To define the right banner as blank, type: WinRightBanner= leaving the remainder of the line after the equal sign blank. The right banner default setting is: WinRightBanner=&q-name &f-title &f-page-number

10. Save the story.

Example Style Story

Here is an example of a local print style story stored in a queue located in SYSTEM.STYLES:



Other examples are provided in a starter database installed with every new iNEWS system.

```
WinWYSIWYG=yes
WinBanner=yes
WinLeftBanner=&q-name PAGE-&f-page-number SLUG-&f-title
WinRightBanner=Tal-&f-presenter USER-&u-name &date &time
WinBlanks=2
WinSegmentPageBreak=yes
WinColorPrint=no
WinScriptShift=no
WinScriptLeftWidth=3.5
WinScriptRightWidth=3.5
WinLeftMargin=0.75
WinRightMargin=0.75
WinTopMargin=1.0
WinBottomMargin=1.0
WinSeparator=yes
WinSpacing=1.5
```



```
WinForm=All ;NOTE-Other options=Yes, No
WinPrintForm=Rundown
WinPageHeader=header.default
WinPageFooter=footer.default
WinHeaderFontFace=Times New Roman
WinHeaderFontSize=10
WinBannerFontFace=Arial
WinBannerFontSize=12
WinFormFontFace=Arial
WinFormFontSize=10
WinBodyFontFace=Times New Roman
WinBodyFontSize=20
WinOrientation=portrait;Note-Only other options=landscape
WinRundownFontFace=Arial
WinRundownFontSize=10
WinRundownLeftMargin=0.50
WinRundownRightMargin=0.50
WinRundownTopMargin=0.75
WinRundownBottomMargin=0.75
WinBannerBlanks=2
WinPrintRundownForm=Rundown
WinPrintHide=yes
WinPrintHideStart=<<
WinPrintHideStop=>>
WinVTEmulation=no
WinCopyDriverSettings=yes
WinSequentialNumbering=no
WinScriptSplit=no
```


12 Wires

The profile files contain information about how the iNEWS system will use the hardware, or instructions on how the software is to behave. Therefore, each wire service available to your system can have a *wire profile file*. This file contains settings for options you can use with that wire service to influence the behavior of the wire. When you restart a wire service, your system checks its profile for necessary information.

This chapter contains the following main sections:

- [Adding a Wire – Avid Data Receiver](#)
 - [Phase 1](#)
 - [Phase 2](#)
 - [Phase 3](#)
 - [Phase 4](#)
- [Wire Profile Options](#)
- [Wire Distribution](#)
- [Setting Up Wire Keyword Searches](#)
 - [Additional Information about Search Jobs](#)
 - [Keyword Search Rule Sets](#)
 - [Keyword Checker Messages](#)

Adding a Wire – Avid Data Receiver

The Data Receiver is a highly configurable wire server designed to handle a newsroom's subscription wire feeds (8 or fewer) in a single server, while giving iNEWS users access to the data from their desktops. It processes incoming wire data from multiple input sources, including e-mail and traditional serial feeds in multiple data formats. For more information on the Avid Data Receiver, see the *Avid Data Receiver Installation and Operations Manual*.

When adding a wire for ingesting through the Data Receiver, you must complete four phases in the setup.

- Phase 1 - Ensure the Avid Data Receiver software is installed, configured, licensed, and the wire feed is connected to the appropriate port.
- Phase 2 - Create an entry for the wire service in the configuration file and a wire profile for the wire service.
- Phase 3 - Add the wire distribution information.
- Phase 4 - Reconfigure the system and start the Avid Data Receiver so it can begin receiving and distributing wire stories.

Phase 1

The procedures for installing and configuring the Data Receiver are provided in the *Avid Data Receiver Installation and Operations Manual*. For integration with iNEWS, additional tasks must be completed.

To ensure proper configuration and licensing, do the following on the iNEWS Server:

1. Add a wire server resource to the iNEWS Server license. An example of a license is shown in [“Licensing iNEWS Components” on page 247](#). Each wire instance on the Data Receiver requires its own wire server resource be licensed and configured on the iNEWS Server.



Configuring more wire server resources in the iNEWS configuration file (/site/config) than are licensed will result in the system failing to configure. If this happens, the system will display a message at the console indicating that license limits were exceeded and by how much.

2. Choose a device number for each Data Receiver instance. This number is required for later phases in the setup. Ensure that the number you choose is not already in use.
3. Connect the wire feed to the selected port on the PC running the Data Receiver software.

Phase 2

Wire service information are added to two files in the Site directory on the iNEWS Server(s):

- The configuration file (/site/config)
- (Optional) The wire profile, located in /site/wires directory

To add wire service information to the configuration file:

1. Select all servers.
2. Modify the iNEWS configuration file by adding the following:



See *“Editing the Configuration File” on page 252 for more information.*

- a. Add a resource list entry to the host section of the /site/config file on the iNEWS Server.

The format is: `reslist <device #> ;<comments>`

For instance: `reslist 801 ;Data Receiver instance`



*For dual or triple server systems, the configuration file will have multiple host sections to define which server handles which devices under various circumstances. Resource list entries should be added to each host section depending on which server you want the Data Receiver instances to run. See *“Configuration File” on page 249 for more information.**

- b. Add a wireserver resource line for each Data Receiver instance to the devices sections of the /site/config file on the iNEWS Server.

The format is: `wireserver <device #> news <source> <join option>`

| Parameter | Description |
|-------------|--|
| device # | The Data Receiver instance device number. |
| source | The wire’s source name (capitalized portion of the distribution name—AP, UP, and so forth). It is the same as the source name used in the Data Receiver. |
| join option | Links stories that have the same title. It is used with wire services that routinely break stories into multiple takes. If you place <code>join</code> in this position, users can view all takes of a story with the Get All Versions workstation option. To leave this option out of the configuration line, enter a hyphen (-). |

Example: `wireserver 801 news AP -`

Adding a Wire Profile

The wire profiles associated with Data Receiver, and located in the `/site/wires` directory, use only two wire profile options—`form` and `idle`. The wire profile must have the same name as the Data Receiver instance’s device number. See [“Wire Profile Options” on page 296](#) for more information.

To add a wire profile:

1. Select all servers.
2. Use the `ed` command in the following format: `ed /site/wires/<device#>`

For instance, type: `NRCS-A# ed /site/wires/801`

Since the file doesn’t exist, the system will create it, but display a line similar to the following: `/site/wires/801: No such file or directory`



This procedure, which modifies the wire profile in the `/site/wires` directory uses `ed`, the line editor. If you do not know how to use `ed`, please see [“The Line Editor, `ed`” on page 655](#).

3. Add the necessary text to the file by typing: **a**

Everything you type after pressing Enter will now be added to the file. The two wire profile options that must be defined for a Data Receiver are:

```
form <name>
```

```
idle <timeout> <interval> <notify group>
```

For instance, type:

```
form wires
```

```
idle 15:00 00:01 inews
```

For more information, see [“Wire Profile Options” on page 296](#).



The words dictionary (`/site/dict/words`) contains `W_WIRE_FORM /wires` which is used as the default if no `form` option is specified in the profile or if the profile doesn’t exist.

4. Stop adding lines to the file by typing a period (`.`) and press Enter.
5. Type **w** to write (save) your changes to disk.



Do not use an uppercase `W` in this step.

6. Type **q** to quit the line editor.

Phase 3

Each Data Receiver instance must have its own set of instructions in a story, called the wire distribution story, located on the iNEWS system in the Distribution queue in the SYSTEM.WIRES directory. When you add a new wire to your system, you need to add new instructions to your system's wire distribution story so it knows how to distribute stories it receives from that wire service.

You can view the story by logging in as a system administrator at an iNEWS Workstation and navigating to SYSTEM.WIRES.DISTRIBUTION, which is the default location for distribution stories.



Avid recommends that you make a backup copy of the story before making any modifications to the wire distribution story, .

To modify the wire distribution story:

1. At an iNEWS Workstation, select the story in the Queue panel. The story's text will appear in the Story panel.
2. Edit the story by changing or adding lines as necessary in the Story panel.
3. Save the story by doing one of the following:
 - a. Click the Save button.
 - b. Select File > Save.
 - c. Press Ctrl+S.



The wire distribution story is typically set up by Avid Customer Support technicians when your system is installed. If you need assistance in modifying an existing wire distribution story, contact Avid Customer Support. Also, a standard copy of a wire distribution story is available in "System Files" on page 541.

Here is a partial sample of a wire distribution story:

```
AP#bloom## wires.bloomberg
AP#kip01## wires.kipling
APb###aa## wires.features
AP#####o# wires.weather
AP#####s# wires.sports
AP#####i# wires.world                URGENT
AP#####w# wires.washington           URGENT
AP#####n# wires.national             URGENT
AP#####a# wires.national             URGENT
AP#####f# wires.biz                  URGENT
AP#####   wires.all                   always  URGENT
```

Phase 4

The final phase in adding a wire is to check your setup by reconfiguring the system and starting the Data Receiver. This will incorporate the changes you made to the configuration file in Phase 2.

To reconfigure the system:

1. Take the system offline by typing: `NRCS-A$ offline`
2. Reconfigure the system by typing: `NRCS-A$ configure`
3. When the prompt reappears, bring the system back online by typing: `NRCS-A$ online`



For information on how to start the Data Receiver, see the [Avid Data Receiver Installation and Operations Manual](#)

Wire Profile Options

Each wire option available may appear only once in a wire profile and may be assigned only one value. Only two options—`form` and `idle`—are necessary in a wire profile for wires ingested through Data Receiver.

- `form <name>`

The `form` option is the form name assigned to the wire story, generally used to view the story. This form is the story form assigned to queues that typically contain wire stories. In wire profiles for Data Receiver instances, the `<name>` variable is usually `wires`.



The words dictionary (/site/dict/words) contains W_WIRE_FORM /wires which is used as the default if no form option is specified in the profile.

- `idle <timeout> <interval> <notify group>`

The idle option controls how your system handles a prolonged period of inactivity, when no stories are received by iNEWS, for a wire. The timeout value indicates how long the wire should be idle before users are notified. For instance, a time-out value of 2:00 triggers notification if no stories are received from this wire in two hours.

The notify group is optional; it contains the user group name on iNEWS that is notified, through the bottom-of-screen message, if the time-out value is reached. A message is always sent to the console when this happens. To disable sending a message to a group, put a hyphen (-) in this position.

The interval value is used to repeat this notification if the wire remains idle for an additional period of time beyond the time-out value. A time-out value of 2:00 combined with an interval of 15 notifies users after two hours of idle time and every 15 minutes thereafter, as long as the wire remains idle.

The time-out and interval can have any values between 2 minutes and 24 hours, expressed in hh:mm format. Providing a value of 0 for the time-out, or omitting the idle option from the profile altogether, disables the idle time alert for this wire. An interval of 0 limits notification to one initial message.

Wire Distribution

The Data Receiver routes each wire story it receives to an appropriate queue, based on which wire service sent the story and the story's distribution code. This distribution system ensures all sports-related wire stories are sent to the sports wire queue, weather forecasts are sent to the weather wire queue, and so on.

This section explains the method wire programs use to categorize and distribute wire stories, and provides a process for customizing the distribution of wire stories to keep up with the changing needs of your newsroom.

The Wire Distribution Story

Each wire must have its own entry in the wire distribution story. The wire distribution story is the first story in the SYSTEM.WIRES.DISTRIBUTION queue. If there is more than one story in that queue, the first story is used. This entry consists of up to 150 instructions that tell the wire how to distribute wire stories, based on their wire service codes.

Each line in the story is called a distribution line. Each distribution line consists of a distribution name, a destination queue, and an optional notification priority constraint—all of which are explained in more detail later in this section.

Although your distribution story was set up for you when your system was installed, you might need to modify it by changing an existing distribution line or adding a new distribution line to the story.



When you add a new distribution line, you must add it above any distribution lines for WIRES.UNKNOWN or WIRES.ALL within the section of the distribution story for that wire.

Here is a sample of a wire distribution story:

```
AP#####a# wires.us-national
AP#####b# wires.us-national
AP#####d# wires.features
AP#####e# wires.briefs
AP#####f# wires.features
AP#####h# wires.business
AP#####h# wires.briefs
AP#####i# wires.us-national
AP#####l# wires.features
AP#####m# wires.business
AP#####o# wires.weather
AP#####p# wires.us-national
AP#####q# wires.sports-score
AP#####r# wires.rundowns
AP#####s# wires.sports-stories
AP#####t# wires.rundowns
AP#####v# wires.advisory.other
AP#####w# wires.washington
AP##### wires.all
AP##### wires.unknown
```

Distribution Name

Each distribution line must begin with a distribution name that sets the criteria a story must meet to be sent to the destination queue specified in that instruction. Begin the distribution name with the wire's source name, as specified in the configuration file. Otherwise, the wire ignores that instruction.

The source name is the contents of the fourth column for the wire in the /site/config file. For instance, the following wire is named AX:

```
wireserver 801 news AX -
```

Follow the source name with the wire service code you want that instruction to handle. There must be no spaces between the source name and code. Within the code, specify the category, service level, selector, and priority codes a story must have for the wire to send it to the destination queue specified in the instruction.

In most cases, you need only specify the category code to distribute wires. You generally use only the source name and category code positions in the distribution name and fill the remainder of positions in the distribution name with pound sign (#) characters, which the wire ignores.

For instance, to create a distribution name that matches all Associated Press sports stories you receive, begin the distribution name with the source name of the wire (AP in this example) and put s in the ninth (second-to-last) position. It does not matter what is in the remainder of the distribution name, so fill those positions with # characters: AP#####s#.



When you enter a distribution name, type the wire program's source name in the same case as it is defined in the configuration file. Similarly, Avid recommends entering wire category and priority codes in lowercase in your distribution names.

Destination Queue

Follow the distribution name with the full pathname of the destination queue where you want the wire to put any stories that match the distribution name. For instance, to have the wire send every story that matches AP#####s# to WIRES.SPORTS, specify that queue:

```
AP#####s# WIRES.SPORTS
```

If you create a distribution line with a destination queue that does not exist, the system creates the queue when you save the distribution story, only if all higher-level directories already exist. For example, to distribute sports scores to WIRES.SPORTS.SCORES queue, the WIRES.SPORTS directory must already exist in the system; if it doesn't, the system cannot create the queue. A queue created this way inherits directory traits assigned to the Wires directory, including the purge interval. If you want a different purge interval, use the Directory Properties dialog box to change it. See [“Changing Database Traits” on page 136](#) and [“Database Purge Intervals and Limits” on page 139](#) for more information.

Notification Priority

Normally, when your system receives a story the wire service has marked as either BULLETIN or URGENT, it notifies everyone logged in. You can change the priority level a story must have to merit notification by setting an optional notification priority constraint in the distribution line.

For instance, to have the wire that handles the AP wire service put a copy of each Associated Press brief it receives in WIRES.BRIEFS and notify you whenever it puts an AP brief marked urgent in this queue, add the distribution line shown here:

```
AP#####d# wires.briefs URGENT
```

You can set four notification priority constraint levels in each distribution line. Each level restricts user notification to a higher degree. By convention, type them in uppercase.

| Priority Level | Description |
|----------------|--|
| URGENT | The lowest constraint level. Any story marked FLASH, BULLETIN, or URGENT by the wire service causes users to be notified if this constraint level is specified. |
| BULLETIN | The next-lowest constraint level. Any story marked flash or bulletin causes users to be notified if this constraint level is specified. BULLETIN is the default priority level; if you do not specify a notification priority constraint in a distribution line, any story that matches the distribution name in that line and has this level or higher priority causes notification. |
| FLASH | The second-highest constraint level. When you set a distribution line to this constraint level, stories that match the distribution line's distribution name must have a priority of "flash" in your system to notify users. |
| SILENT | The highest constraint level. It prevents your system from notifying users of the arrival of any story processed by that distribution line. No story ever has a high enough priority to override this constraint level. |

WIRES.ALL Notification Priority

Your system sends only one notification for a story, even if the story matches two or more distribution lines. However, since each line might have a different notification level, the system chooses the lowest notification level and uses it to determine whether to send a notification.

This is important when considering the distribution line for WIRES.ALL. By the way it is typically set up, each story received is distributed to this line as well as others. Consequently, the notification priority set for this line can also affect how notification levels are interpreted for other lines.

For instance, if the distribution line for WIRES.ALL has a notification level of BULLETIN, any other distribution line set to a notification level of FLASH or SILENT (higher levels) is downgraded to BULLETIN.

However, by setting the distribution line for WIRES.ALL to SILENT (the highest notification level), you can make certain that all other distribution lines for that wire have notification levels lower than WIRES.ALL. This ensures their notification levels take precedence over those for WIRES.ALL.

Transmit or Always Options

You can apply two other options, TRANSMIT and ALWAYS, to a distribution line.

| Option | Description |
|----------|--|
| TRANSMIT | This option is used on distribution lines that send stories to queues scanned by distribution servers. It causes the wire to give the story a distribution code that matches its wire service code before sending it to the destination queue. This way, a distribution server that scans the destination queue can distribute the story, based on its original wire-service code. |
| ALWAYS | If you want the wire to distribute a copy of every story it receives to a certain queue, place ALWAYS on the distribution line that sends stories to that queue. The word ALWAYS may follow or precede a notification priority constraint. Most often, this is used to send a copy of every story received by the AP wire service to WIRES.ALL: <code>AP##### wires.all ALWAYS</code> |

WIRES.UNKNOWN

A wire puts stories that do not match any of its distribution lines in the queue WIRES.UNKNOWN. Most systems have a distribution line that uses a wildcard distribution code and the ALWAYS option to send a copy of every story to WIRES.ALL. As a result, no story is unknown as far as the wire is concerned.

In many cases, you want to send stories that do not match normal distribution codes to both WIRES.ALL and WIRES.UNKNOWN. To do this, you need a special distribution line in the wire distribution story that sends a copy of any wire that does not match a normal distribution code to WIRES.UNKNOWN before sending it to WIRES.ALL.

If your system uses more than one wire service, the distribution lines for each service should be in separate groups. Each group should have its own distribution line for WIRES.UNKNOWN at the bottom of that group of distribution lines, just above that group's distribution line for WIRES.ALL.

Maximum Number of Lines

Each wire's entry in the distribution story can contain up to 150 distribution lines. To fix a distribution story so your system can use it, divert distribution wires to the distribution server so the total for each wire is no more than 150. Once you remove excess distribution lines and save the story, your system should accept the distribution story and begin using it.

Avoiding Hidden Categories

When you add lines to your wire distribution story, you might get hidden category error messages when you save the story. These errors occur when the line or lines you have added are not in the proper position relative to other lines in the story. The following example illustrates this problem and shows how to fix it.

Suppose you had this line in your wire distribution story:

```
UP#####g#      wires.regional
```

In addition to this line, you wanted to add a line like this one:

```
UP#xyz##g#      wires.briefs
```

While this may seem straightforward, you get a hidden category error message if you add the new line following the existing one. To understand why, look at the two lines together.

```
UP#####g#      wires.regional
```

```
UP#xyz##g#      wires.briefs
```

Suppose a story came from the wire service with a distribution code of UPoxyzoogr or something else that would ordinarily match your new WIRES.BRIEFS distribution line. The problem arises because the story also matches the WIRES.REGIONAL line above that one, and, since the wire distribution program stops checking the distribution story after it finds a match, it never gets to the WIRES.BRIEFS line.

To avoid hidden category errors:

- ▶ Ensure that you arrange any distribution lines that have common elements so the more specific line comes first in the distribution story, like this:

```
UP#xyz##g#      wires.briefs
```

```
UP#####g#      wires.regional
```



This rule of distribution does not apply to distribution lines that have the same code but distribute the stories to different queues. These lines still match the appropriate wire stories, as long as they are placed together in the distribution story.

Mailboxes

Your system was installed with the reserved distribution mailbox assigned to the queue SYSTEM.WIRES.DISTRIBUTION. This mailbox allows the wire distribution checker (a server program) to know when you make changes to the distribution story. The checker in turn notifies the wire programs that the distribution story has changed. This is the only way the wire program knows to incorporate changes you make. See [“Mailbox Tasks” on page 325](#) for more information.



Do not assign a different mailbox to this queue or assign this mailbox to any other queue.

If you want everyone logged in on the system to be notified when a wire story is placed in WIRES.ADVISORY.PRIORITY, you must assign the reserved mailbox “all” to that queue. Mailboxes are assigned to queues using the Queue Properties dialog box. See [“Changing Database Traits” on page 136](#) and [“Reserved Mailboxes” on page 327](#) for more information.

Purge Intervals

Nowhere in your database are proper purge intervals as important as in the Wires directory. All queues in the Wires directory should have roughly similar purge intervals. If not, the system may not reclaim space used by old wire stories efficiently.



Never use the wire distribution story to distribute wire stories to queues not purged. Doing so can eventually lead to a low-on-space condition as the queues fill up with old stories.

Internationalization

You can modify the FLASH, URGENT, BULLETIN, SILENT, ALWAYS, and TRANSMIT options by changing appropriate entries in the words dictionary (/site/dict/words).

You must use the maketab command for the system to recognize dictionary changes.

The system expects each option to begin with a unique letter and looks at the first letter of the word to determine what it is. The system would find TRANSMIT to be the same as TX.

Setting Up Wire Keyword Searches

In addition to receiving and distributing wire stories, a wire can also search stories it processes for keywords you want. When it finds a story that satisfies one of your search rules, it copies that story to a queue you specify. Using wire keyword searching can be an easy way to collect stories of interest to specific writers in your newsroom.

Utility programs, called *keyword servers*, can do keyword searching in the same way as wires. For example, they can perform this task on queues to sort scripts for archiving. They are listed differently in the search job list, but otherwise searching and story distribution are handled the same way.



See [“Adding a Keyword Server” on page 361](#) for more information.

To set up a keyword search for a wire or keyword server:

1. Check the search job list for existing entries.

The search job list is the first story in the SYSTEM.WIRES.KEYWORDS queue.

Each job list entry uses the following format:

```
*[<number of queues>] [<name>] [< name>]
[<queue name>] [<base distribution folder>]
. . .
[<queue name>]
```

Begin each entry with an asterisk (*).

Follow the asterisk and number combination with the device name of the program that uses this entry or a source name for a wire. List two or more names if you want more than one program to use the entry. Program names are specified in the configuration file.

Follow the first line with one or more lines that specify queues containing keyword stories you want the program to use. A keyword server can have as many as 20 queues.

If the program that uses the entry is a wire program, you can list as many as five different queues here.

2. Create a keyword queue.

By convention, keyword stories are placed in queues in SYSTEM.WIRES, so create a queue called SYSTEM.WIRES.KEYWORDS-AP.

See [“Creating a New Queue” on page 104](#) for more information.

3. Assign the reserved keyword mailbox to the new keyword queue.

If this mailbox is not assigned to this queue, your system cannot notify the keyword server when you add new search rules to the keyword story or modify existing ones. The keyword server will not examine your changes for mistakes or notify programs that use the keyword story of changes you made.

See [“Keyword Mailbox” on page 311](#), [“Changing Database Traits” on page 136](#) and [“Reserved Mailboxes” on page 327](#) for more information.

4. Add an entry to the search job list.

Ensure the keyword story queue you just created is listed in the wire program's search job list. The iNEWS system looks for the keyword story in the SYSTEM.WIRES.KEYWORDS-AP queue, so the wire program's search job list should look similar to this:

```
* AP
system.wires.keywords-ap
;
* kwdserver1
system.keywords.shuttle
system.keywords.recycle
```

When you save the story, the keyword checker examines your changes for errors. Then it notifies all Data Receiver wire services that keyword changes were made.



The keyword checker (kwdcheck) validates keyword search stories just as the distribution checker (catcheck) validates wire distribution stories. The keyword checker is different from the keyword server, which is the utility program doing the actual wire keyword searching.

The number of queues you specify determines how many unique keywords can be in each story. The formula for this is the maximum number of unique keywords a wire program can use (250) divided by the number of stories you declare. For instance, listing two queues would mean each story could have a maximum of 125 unique keywords. See [“Additional Information about Search Jobs” on page 306](#) for more information.

5. Create a keyword story.

Wire and keyword server programs use the first story in the queue as the keyword story. If there is more than one story, ensure your keyword story is at the top of the queue.

6. Add search rules to the keyword story. See [“Keyword Search Rule Sets” on page 307](#) for more information.

7. Save the keyword story.

The keyword checker examines the story for errors and creates any destination queues that do not exist. If the keyword checker finds an error, it sends a message to your workstation.

The most important of these messages advises you the story contains a destination queue that does not have a purge interval.

Using a non-purging destination queue does not prevent the keyword story from being used, but you risk collecting stories in that queue until the system runs out of space. Avid strongly recommends all your destination queues have purge intervals.

The messages you see after saving a keyword story are advisory messages and do not prevent the keyword story from being used. Once you have viewed all the messages, the program for which the keyword story was designed begins using the rule sets in that keyword story to search the stories it processes, as long as an entry for that program has been made in the search job list.

Additional Information about Search Jobs

The following section provides additional information about search job list features and important considerations.

Suppressing a Search

To prevent a program from doing any searching, create an entry for the program that consists of an asterisk only and the program's device or source name, or do not include an entry for the program in the search job list. If your search job list contains a default entry, you need to include entries for each wire and keyword server program you want to keep from doing any searching.

Default Entry

A default entry is one that begins with an asterisk, but does not specify a device or source name. A search job list must contain a default entry located at the bottom of the story. If a program is unable to find an entry that contains its device name, it uses the default entry.

Your system uses a special utility (server) program, called *keyword checker*, to check any changes you make to the search job list. For the system to notify the keyword checker when you modify the search job list, assign the reserved system mailbox keyword to the SYSTEM.WIRES.KEYWORDS queue. See [“Changing Database Traits” on page 136](#) and the [“Maintain Tab” on page 130](#) for more information.

Keyword Limitations

Each wire can have search rules containing no more than 250 unique keywords. Divide the search rules among as many as five keyword stories in different queues. The maximum number of unique keywords you can use in each queue's keyword story is 250 divided by the number of queues you specify in the program's search job list entry.

In a similar way, keyword servers can use search rules containing a total of 1000 unique keywords. Divide these search rules among as many as 20 different queues. The number of unique keywords allowed in each keyword story is equal to the total number allowed for that keyword server divided by the number of queues you include in the keyword server's search job list entry.

Keyword Search Rule Sets

Each keyword story is a collection of rule sets. Each rule set contains the search rules you want your program to use to identify stories relating to a particular topic.

The format each rule set must have is as follows:

```
destination <queue name>

<search rule>

[<search rule>]

. . .

[<search rule>]
```

Each rule set begins with a destination line naming the queue where copies of stories that satisfy any of the rule set's search rules are placed. This line must begin with the word `destination`, followed by the full pathname of the destination queue or the default directory path. See [“Default Directory Path” on page 310](#) for more information.

Two rule sets cannot use the same destination queue.

Follow the destination line with one or more lines that contain search rules you want a particular wire and keyword server program to use to search. Each search rule occupies a separate line in the rule set.

For instance, the rule set in the following example begins with a destination line that tells the program to put copies of stories that satisfy either of the rule set's search rules into `WIRES.KEYWORDS.DRUGS`. The two lines that follow contain the search rules the program uses to find stories relating to drugs:

```
destination  wires.keywords.drugs

    panama AND drugs

    drugs NOT pharmacy OR (drugs AND smuggling)
```



Never create a destination line that sends stories to a personal queue or any queue not purged. Having a program send stories to such queues can lead to an out-of-space condition.

Each search rule consists of one or more keywords you want a wire or keyword server program to use to locate stories on a particular topic.

Each search rule is contained in a paragraph and is limited to approximately 1000 characters or to the end of the paragraph, whichever is less. You can have more than one search rule defined for each destination queue. The only limitation is that you cannot exceed the number of unique keywords allowed for the story. For details on the maximum number of unique keywords, see [“Keyword Limitations” on page 306](#).

If your search rule contains more than one keyword:

- Include a space before and after each keyword so the system can identify the start and end of each keyword. You can change default delimiters by customizing the dictionary translation for the W_DELIMITERS token in the words dictionary. Use the maketab command to let the system recognize dictionary changes. Your iNEWS system must have all users logged out and all devices stopped to use the maketab command.

Delimiters defined by W_DELIMITERS are used by the wire, seek, and keyword server programs to find the start and end of words in stories. These isolated words are matched with keywords from the search list. Definition of the delimiters is not related to the task of constructing search rules. Keywords are limited to a maximum of ten characters.

- Include the AND, OR, or NOT operator between two keywords:

AND requires that a story contain multiple keywords. For instance, a rule like “drugs AND smuggling” finds only stories that contains both of these words.

OR requires that a story have any of multiple keywords. For instance, a rule like “cnn OR panama” finds any story that contains either of these words.

NOT specifies that a story must not contain the keyword following the operator. For instance, a rule like “drugs NOT pharmacy” finds any story that has the word “drugs” in it but does not contain the word “pharmacy.”

Using Parentheses in Search Rules

Wire and keyword server programs evaluate a search rule from left to right. For instance, to find stories that contain either “air quality,” “clean air,” or “dirty air,” create the following long search rule:

```
air AND quality OR air AND dirty OR air AND clean
```

Because these programs evaluate rules from left to right, they first see if a story contains both “air” and “quality.” If a story does not contain both of these keywords, the program checks whether it contains both “air” and “dirty.” If the story does not, the program checks whether the story contains both “air” and “clean.”

Using parentheses, you can control how a program evaluates the rule. For instance, by rearranging the rule, we greatly simplify it:

```
air AND (quality OR dirty OR clean)
```

You can use as many sets of parentheses as necessary to have the program evaluate the search rule.

For instance, to create a search rule that finds any story that contains either the word “smog” or the words “air” and “quality,” but not “pollen” or “airline,” build a rule such as the following:

```
smog OR ((air AND quality) NOT (pollen OR airline))
```

After you add your search rules to the keyword story and save, the keyword checker examines your changes for errors. If there are no errors, it notifies any program that uses that keyword story of what you have done so the program can use your changes.

Tips on Building Search Rules

- Each keyword must be a single word; two words without an operator between them generate an error. For instance, you must use space AND shuttle rather than space shuttle to find stories about the space shuttle.
- Each search rule is contained in a paragraph and is limited to approximately 1000 characters or to the end of the paragraph, whichever is less. Searches are not case-sensitive, so you can use uppercase or lowercase letters.
- Be careful using keywords that form the first part of larger words. For instance, “air” finds stories that contain words like “airline” and “airports” as well as “air.” To use a keyword like “air,” include the NOT operator to exclude other words that begin with that keyword, such as NOT airline.

However, wire and keyword server programs do not find keywords when they appear in the middle or at the end of a word. For instance, searching for “ports” does not locate stories that contain words like “airports” or “sports.”

The success of your search depends on how well you design the search rules; they should be specific enough so programs that use them find only stories that relate to your topic.

Scanning stops either at the end of the story or after a maximum of 50 keywords are found. Then the search rules are used on the set of located keywords. The program abandons that rule set when it finds a search rule the story satisfies or when it exhausts all search rules in the rule set.



You can increase the efficiency of your searches by putting rules that are most likely to find stories at the top of each rule set. Put the least complex rules at the top.

User Notification

You can have the system notify you when a program finds a story that contains one of the keywords for which you are searching. Assign a group you are in to the notification group trait of the queue(s) where the program sends the stories. See [“Notification Group” on page 172](#) for more information. Your system then notifies you every time the program places a story in one of these queues.

Removing a Rule Set

When a rule set becomes obsolete, remove it. Because a large number of keyword searches slows your system, removing unused rule sets helps maintain system efficiency.

If you are no longer using the rule set’s destination queue, remove it. This frees up space in the database and helps keep it from being cluttered up with unused queues.

Sending a Story to More Than One Queue

Wire and keyword server programs check stories they process against each rule set in the keyword stories they use. When the program finds a match, it sends a copy of the story to the rule set’s destination queue and continues searching, using the remaining rule sets. Only the first 50 words identified in the story will be used when checking the rule sets.

You can have a story copied to more than one destination queue by repeating the same search rule in different rule sets.

You can also distribute to more than one queue by using a temp queue and an action server. Have your keyword server send the story to the temp queue. This is a queue where the story will only sit for a moment. It is typically hidden from ordinary users. Assign an action server mailbox to this queue. In the action server joblist, have it scan the queue and dup the story to each of the destination queues. On the last list, have the action server move rather than dup. That will prevent stories accumulating in the temp queue. The temp queue should have a long purge interval, to protect against story loss if the action server fails. Restarting the action server in that instance should result in all accumulated stories being distributed.

Default Directory Path

You usually need the full pathname to specify each rule set’s destination queue. You must do this for the program performing the search, to locate the destination queue.

Sometimes, all rule sets in a keyword story contain destination queues in the same directory. In this case, set a default directory path for the keyword story and then specify the destination queues by name rather than using the full pathnames.

To specify a default directory, enter the queue's default directory in the search job list entry that uses the keyword story. Follow the queue name that contains the keyword story with the default directory pathname. For example set the keyword up like this:

```
* AX
```

```
system.wires.keywords-apx      wires.keywords
```

Then, in SYSTEM.WIRES.KEYWORDS-APX, the entries can look like this:

```
destination      cancer
                  cancer
```

The actual destination is WIRES.KEYWORDS.CANCER.

Use this feature with group security. You can allow users to manage their own keyword searches but control where they send the stories.

Keyword Mailbox

Assign the reserved keyword mailbox to the queue SYSTEM.WIRES.KEYWORDS, which contains the search job list. Otherwise, your system cannot notify the keyword checker when you make changes to the search job list.

Always assign the keyword mailbox to queues that hold your keyword stories. If this reserved system mailbox is not assigned, your system cannot notify the keyword checker when you create new search rules or modify existing ones.

When you modify an existing search rule or create a new one, the keyword checker alerts programs that use that rule about the change. If the queue does not have the keyword mailbox assigned to it, the keyword checker cannot do this. Consequently, programs that use the search rule you created or modified will not know about your changes until you restart them.

See [“Changing Database Traits” on page 136](#) and [“Reserved Mailboxes” on page 327](#) for more information.

Keyword Checker Messages

The following messages from the message dictionary may appear when you modify and save a keyword story:

| Message | (Token) | Description |
|-----------------------------------|--------------|---|
| Could not enter the queue | (M_BADQUEUE) | The specified destination is not accessible or an error occurred while creating or entering the queue. |
| Default keyword list must be last | (M_WNOTLAST) | Placing the default keyword list anywhere but at the bottom of the story will prevent those keyword lists that follow it from being used. |
| Duplicate destination | (M_DUPEDEST) | The specified destination has already been used for another keyword list. |
| Keyword too long | (M_KWDLONG) | Length of the keyword string is longer than maximum length allowed. The maximum is 10 characters. |
| Maximum file number bad | (M_FILENUM) | <p>The number of keyword stories has exceeded the maximum allowed.</p> <p>The M_FILENUM diagnostic is produced if there is anything other than a space, a five, or a twenty following the asterisk in the * AP line of the wire program's search job list.</p> <p>The * AP is equivalent to *5 AP meaning that you can have at most 5 queue names specified for AP wire search rules. Wires are limited to 250 unique keywords in the search rules. So you can have 5 stories with each story containing 50 unique keywords. If you have one story, it can contain 250. If you have 2 stories, each story can contain 125, and so forth.</p> <p>Keyword servers are limited to 1000 unique keywords in the search rules. You can specify *20 kwdserver1 to indicate that this keyword server may have as many as 20 queue names specified that will contain the search rules for it. The example on page 395 limits the list to 5 and therefore at most 250 unique keywords for kwdserver1.</p> |
| missing | (M_MISSING) | The keyword story does not understand the specified rule set. Either an operator is missing (AND, OR, or NOT) or an unmatched parenthesis exists. |
| No destination found | (M_NODEST) | No destination is specified for a given rule set. |
| Not a queue | (M_NOTQUEUE) | The specified destination is not a valid queue name. Check whether it is misspelled, or whether it is a directory. |

| Message (Continued) | (Token) | Description (Continued) |
|-------------------------------------|----------------|---|
| Queue is never purged | (M_PURGEZERO) | The specified destination queue is never purged, which means stories may accumulate in it and eventually cause a space problem in the database. |
| Syntax error | (M_SYNEROR) | The keyword checker does not understand the specified line. Check whether format of the line is correct. |
| System error | (M_SYSERROR) | A system error has occurred that caused the keyword checker to terminate. If you get this message, call Avid. |
| Too many expressions | (M_EXPMAX) | A single search rule is too complicated. It contains more than 500 words/operators. |
| Too many keyword distribution files | (M_KWFMAX) | The number of keyword distribution files exceeded the maximum allowed. The max is 5 for wireservers and 20 for keyword servers. |
| Too many keywords | (M_KWDMAX) | The total number of unique keywords processed by the keyword checker has exceeded the maximum allowed. |
| unexpected | (M_UNEXPCTD) | The specified line has extra punctuation, such as an unmatched parenthesis. Ensure that parentheses appear only in pairs. |
| UPDATE queue | (M_UPDATEQ) | The specified destination queue is an update queue. This means that each time a story is placed into the destination queue, the queue must be searched to determine if a copy of the story is already in the queue. This puts an unnecessary load on the system. Destination queues should not have the update trait. |

13 Servers

Do not confuse servers in this chapter with physical servers, which are computers with the iNEWS database and running the iNEWS application software. Those computers are referred to in this chapter as iNEWS Server, server A, server B, and so forth. This chapter explains setup and use of utility programs, also known as servers. There are various types of these server programs, which perform a multitude of tasks.

This chapter contains the following main sections:

- [Overview](#)
- [Adding a Server Program to the System](#)
- [Job Lists: Queues, Stories, and Commands](#)
 - [Types of Tasks for Servers](#)
 - [Mailbox Tasks](#)
 - [Timed-Interval Tasks](#)
- [Action Servers](#)
- [Distribution Servers](#)
- [Parallel Wire Servers](#)
- [Keyword Servers](#)
- [System Servers](#)
 - [Seek Servers](#)
 - [Fast Text Search Servers](#)
 - [Mail Server](#)
- [Monitor Servers](#)
 - [Checklist: Monitor Server Configuration](#)
 - [Creating Styles](#)
 - [Using the Monitor Server](#)
- [Network iNEWS Systems Using RX/TX Links](#)

Overview

Servers are utility programs, which automatically perform various actions on stories in the iNEWS database. Some are general purpose and do jobs you define; others are narrowly defined, needing no instruction from you. You can connect (network) server programs in separate systems together using rx/tx links.

These various types of server programs are explained in more detail in individual sections later in this chapter.

The types of servers available to you are:

- Action servers – general-purpose servers that move, duplicate, remove, or replace stories added to or edited in certain queues
- Distribution servers – distribute stories to different queues, based on each story's distribution code
- Parallel wire servers – allow you to have a wire distribution backup in case you have problems with one of the Data Receivers.
- Keyword servers – perform keyword searches on stories added to or modified in the designated queues and then copy or move matching stories to a queue you specify
- System servers, including:
 - Seek servers - perform searches
 - Fast Text Search (FTS) servers - perform fast text searches and index stories in the FTS database
 - Mail servers - handle mail processing
- Monitor servers – check a show's event requests for errors, create composite and event lists, and send playlists to iNEWS Command or to an external playback system through an iNEWS MOS Gateway.

Adding a Server Program to the System

The overall process of creating a server is the same for all server utility programs.

To add a server to iNEWS:

1. Check the configuration file (`/site/config`) and choose an available device number for the server. Device numbers can range from 1 to 5000. Take note of existing device ranges in your `/site/config` file. There might already be servers of the type you are adding. If possible, add your new server to the end of the number range the existing servers use. When setting number ranges, be sure to allow for expansion.
2. Add the server program to the configuration file on each iNEWS Server—such as server A and server B—in your system.



Always back up the `/site/config` file before making any changes. See “Editing the Configuration File” on page 252 for more information.

- a. Select all iNEWS Servers.
- b. Open and edit the configuration file, by typing (what appears in bold):

```
NRCS-A$ ed /site/config
1259
```

After you press Enter, the editor responds by displaying a numerical value indicating the file size expressed as the number of characters, including spaces and returns.



This procedure, which modifies the `/site/config` file uses `ed`, the line editor. If you do not know how to use `ed` to modify lines in the file, please see “The Line Editor, `ed`” on page 655.

- c. Add the server program’s device number to the servers line in the host section(s) of the configuration file for the iNEWS Server on which the server program will run. For instance: `servers 257 258 259 260`
The device number 260 is added to the servers line in this example.
- d. Add a configuration line for the server program. This line should contain the mailbox number assigned to the server program, when necessary.



Divide your server programs evenly among your iNEWS Servers to distribute the load they put on your system. Additionally, ensure that you modify the servers line for alternate host definitions for your iNEWS Servers to include the server program’s device number. This ensures the server program can run on the surviving computer should one of your iNEWS Servers stop functioning. See “Configuration File” on page 249 for more information.

The format for server programs’ configuration lines are:

```
server <device #> <type> <mailbox> <device name>
```

| Parameter | Description |
|-------------|--|
| device # | The device number assigned to the server program. This number must also be listed in the servers line in a host definition. |
| type | The type of server program, such as action, distribution, keyword, seek, monitor, mailserver, ftsseek, and ftsindex. |
| mailbox | The mailbox the server program uses. Valid standard mailbox numbers are 1 through 5000. This number typically matches the server program's device number. See “Types of Mailboxes” on page 326 for more information. |
| device name | To give the server (utility program) a device name, enter that name here (up to 8 characters). If not, enter a hyphen. |

The following are sample configuration lines for various server programs:

```

server      201      mailserver   201      - ;mailserver
server      211      keyword     211      keyword ;keyword svr
server      231      seek        231      - ;seek server
server      241      ftsindex    241      - ;fts indexing
server      242      ftsseek     242      - ;fts searches
server      251      action      251      - ;action svr
server      251      action      -        - ;timed-action svr
server      271      monitor     271      - ;morning show
server      281      rxnet       -        -
server      291      txnet       291      -

```

Comments appearing after the semicolons (;) are optional.

- e. When you finish making changes to the configuration file, save your changes and exit the line editor by typing what appears in bold:

```

w
1279
q

```

Do not use an uppercase (W) in this step. When you press Enter, the editor responds by displaying a numerical value indicating the file size, which should be identical on all servers. See [“The Line Editor, ed” on page 655](#) for more information.



Information pertaining to steps 3, 4, 5, and 6 vary depending on the type of server program. These steps are described in more detail in individual sections for each server program later in this chapter.

3. Create a job list queue in the System directory of the iNEWS database. See [“Creating a New Queue” on page 104](#) for more information. The name and location of the job list queue in the System directory varies, depending on the type of server program.
4. Create a job list story with defined tasks.



Check your job list before you save the story. When you start the server, it checks for syntax errors in time intervals, but does not catch errors such as misspelled queue names. You must make sure everything is correct.

5. (Optional) Additional queues and stories may be required for some utility programs, such as distribution and parallel servers. See individual sections for each server program later in this chapter for more information.
6. Assign mailboxes to queues, if necessary. The queue’s mailbox number should match the server program’s mailbox number, which appears in its configuration line in the /site/config file.
7. (Optional) Test your configuration changes. See [“Testing the Site Configuration File After Alteration” on page 253](#) for more information.
8. Reconfigure the system. See [“Incorporating Configuration Changes” on page 253](#) for more information. When adding server programs, you do not need to stop anything to reconfigure the system.
9. Start the server program using the following command format: `restart <device #>`
For instance: `NRCS-A$ restart 271`
10. (Optional) Back up site files.

Job Lists: Queues, Stories, and Commands

A server program, such as an action or distribution server, requires a job list queue to hold its job list story. A job list story is a set of instructions telling the server program what you want it to do. Each action, distribution, parallel, and keyword server must have its own job list queue in the SYSTEM.CONFIGURE directory. The job list story must be the first story in the appropriate queue. Also, the job list queue name must start with the server's device number.

For instance, an action server with device number 257 would have a job list queue named 257 in the Configure folder of the System directory. The full pathname would be: SYSTEM.CONFIGURE.257. You may want to add a description of the server's function as part of its queue's name. Do this by using a hyphen, such as:

```
SYSTEM.CONFIGURE.257-DUP-TO-ARCHIVE.
```

Further examples are provided in procedures for adding various server programs in their corresponding sections later in this chapter. A complete list of job list commands is provided in [“Job List Commands” on page 527](#).

Types of Tasks for Servers

The job list story outlines tasks the server program is instructed to execute. There are two types of tasks: mailbox tasks or timed-interval tasks. Most servers execute mailbox tasks, while action servers and tx links can perform either timed-interval tasks or mailbox tasks.

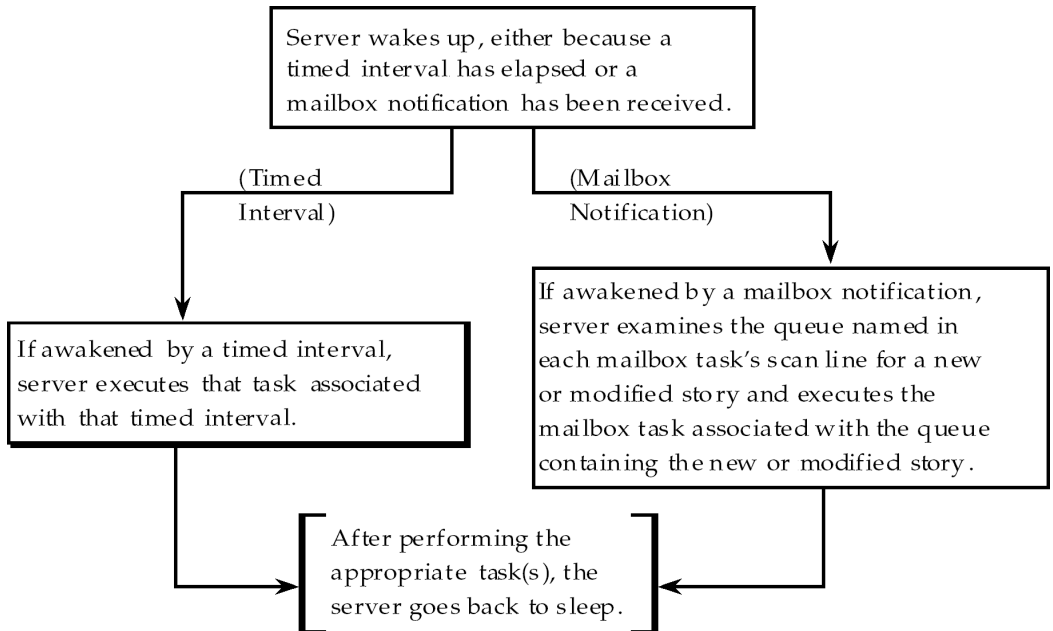
Both tasks must have a scan line identifying the queue with which the task is associated. You cannot specify a directory in an attempt to have the server process all sub-directories and queues. See [“Adding a Scan Line in a Job List Story” on page 321](#) for more information. Follow this scan line with job list commands you want the server to execute if stories have been added to or modified in the task's scan queue. These commands are the task's command set. See [“A Server's Command Set” on page 323](#) for more information.

Mailbox tasks are activities the server program is prompted to do whenever something happens to a queue or its contents. Timed-interval tasks are activities an action server or tx link is instructed to do at certain times.

The server program is inactive and said to be “asleep” until it is “awakened” to execute a job list, either when scheduled to (a timed-interval task) or when stories are added to or changed in a queue (mailbox task).

The difference between the two tasks is that a timed-interval task begins at specified time intervals and a mailbox task does not. Also, timed-interval tasks are only associated with action servers and tx links, while mailbox tasks are used by other utility programs.

The sequence of actions taken by an action server is shown in the following diagram.



Adding a Scan Line in a Job List Story

Both mailbox and timed-interval tasks must have a scan line identifying the queue you want associated with the task. The server can perform commands in the task's command set on any new or modified stories in this queue.



It is also possible and even common to have the server perform tasks on all stories in a queue without regard to their modified status; an example of this is a server that performs auto-archiving tasks.

A scan line must begin with the word scan followed by the full pathname of a queue you want the task to watch.

Its location in the story depends on the type of task. In a mailbox task, the scan line is the first line in the task. See [“Mailbox Tasks” on page 325](#) for more information. In a timed-interval task, the scan line follows the time interval commands. See [“Timed-Interval Tasks” on page 329](#) for more information.

To add a scan line:

1. Open the job list story.
2. Enter the scan line in the appropriate task using the following format:

```
scan <queue name> [all | priority | everyentry]
```



A normal scan reads stories in its scan queue in chronological order, with the oldest first. To reverse this order, use bscan instead of scan. Apart from this distinction, scan and bscan operate identically. Use bscan for long queues. By default a scan only processes modified stories, which are stories added to or edited in the queue. The everyentry option causes different behavior.

If a server's job list tells it to scan more than one queue, each scan only processes 10 stories at a time from a queue before going on to the next queue, unless the queue is designated as a priority queue or the all option is used. The all option overrides the 10 entry limit and will cause all entries in the queue to be processed before the server moves on to the next scan set. The all option differs from the priority queue because the scan queue does not preempt other scan sets and there is no limit to the number of scan sets with the all option specified.

Defining a Priority Queue

You can set up a scan queue as a priority queue for processing new stories. This means that the 10-story limit for scan does not apply to that queue; all qualifying entries in the queue are processed before the server moves on to the next queue. If the priority queue has no stories to process, the system will scan each of the queues in normal order, returning to the priority queue between each queue to check for new stories.

To designate a scan queue as a priority queue:

- ▶ Add priority to the scan line after the queue's pathname.

For instance, type: `scan SHOW.SCRIPTS.AM priority`

Defining an Every Entry Queue

The everyentry option forces the server to process each entry in a queue, not just modified entries.

To designate a scan queue as an every entry queue:

- ▶ Add everyentry (one word) to the scan line after the queue's pathname.

For instance: `scan SHOW.SCRIPTS.AM everyentry`

A Server's Command Set

Follow a task's scan line with a command set containing the commands you want the server to perform on qualifying stories in the queue.

Many servers have unique job list commands only they can use. However, almost every server can use the following five commands: dup, replace, move, remove, and number. Each command and its format are explained in this section, starting with dup.



A complete list of job list commands is provided in “Job List Commands” on page 527. Also, see “Ordered Queues and the Order Command” on page 325.

The format for dup is: `dup <destination queue name> [distribution code]`

This dup command copies stories in the scan queue to a queue you specify, optionally including a distribution code. It must be followed by the name of a queue into which you want the server program to duplicate stories.



When a server, such as an action server, duplicates a story, it replaces any old version of the story in the destination queue with the updated version only when the update database trait is applied to the destination queue. When dup'ing, moving, or replacing a deleted queue entry, server programs will not issue notifications; however, an environment variable (DELETE-NOTIFY) may be used to change this behavior.

The format for replace is: `replace <destination queue name>`

Updates a story in the destination queue only if the original is changed in the scan queue; it does not duplicate new stories from the scan queue into the destination queue. The original story must already exist in the destination queue. Also, the destination queue must have the update database trait.

The format for move is: `move <destination queue name> [distribution code]`

The move command moves stories from the scan queue to a queue you specify, optionally adding a distribution code to them. It must be followed by the name of a queue into which you want the server to move stories and be the last instruction in a command set.



For dup, replace, and move, the destination queue name may be formatted using the strftime syntax to create names based on the current date and time. Use `man strftime` on the console to get a description. The queue will be automatically created if it doesn't exist as long as the beginning of the path already exists.

The format for remove is: `remove`

The remove command deletes the story from a scan queue. It does not send stories to the Dead queue but deletes them immediately. It must be the last instruction in a command set.



Do not include both the remove and move command in a single scan set. The server program cannot do anything else to the story once it executes either one of these commands. Only one of these commands can appear in each command set.

Here is an example of a command set that has an action server duplicate new or modified stories in SHOW.SCRIPTS.AM to BACKUP.MORNING-SCRIPTS and then remove them from SHOW.SCRIPTS.AM:

```
at 10:00
on mon tue wed thu fri
    scan show.scripts.am
    dup backup.morning-scripts
    remove
```

The format for number is: `number <field name> <length> <error queue name>`

Places a unique number in a specified name of each scanned story when the story is moved or duplicated to a new location. The command should include the field name in which the number is placed, the length in digits of the number (from 1 to 5), and the name of an error queue where stories can be placed if the number cannot be inserted in their forms.



The error queue name may be formatted using the strftime syntax and will be automatically created if it doesn't exist.

Depending on how many digits you specify, the story number can range as high as 65535, at which point it resets and starts over at 1. The number is also reset to 1 each day at midnight. Specifying fewer digits results in the number rolling over at 9, 99, 999, or 9999. The number is reset at startup. The version of the story with the number in it does not replace the original story in the scan queue.

The numbered story is used for all operations in the command set that follow the number command. So, if dup comes before number in the command set, the story duped is the story from the source queue. If move or dup comes after number in the command set, the numbered story is used for the move/dup.

Processing Deleted Stories

Ordinarily, servers process stories deleted from their scan queues. This counts as a modification.

To override this function so that the server takes no action when a story is deleted:

- ▶ Include the ignore-del command in the job list, preceding any scan lines you want to ignore deleted stories.

To switch back to the default handling of deleted stories later in the job list:

- ▶ Include the send-del command at that point.

Ordered Queues and the Order Command

Action servers and tx links can order queues with the order command in the job list command set. When dup, move, or replace commands are specified while the order option is enforced, the destination queue will be ordered to mirror the scan queue. The order command can be turned on or off. The format is: `order [yes | no]`

The yes option turns on the order command and the no option turns it off. The order command is reset (turned off) at the start of each new scan set within the job list story.



The order command cannot be placed between open and put commands, which are used by tx links. Otherwise, it can appear anywhere within a scan set to control the ordering of queues. Both open and put commands can appear between order yes and order no commands.

Here is an example of the order command used in an action server's job list story:

```
scan      main.rundown.6pm all      ;all option on
          order                  ;order turned on
          dup slave1.rundown.6pm ;will be ordered
          order no                ;order turned off
          dup archive.rundown.6pm ;not ordered
          order yes                ;order turned on
          dup slave2.rundown.6pm ;will be ordered
scan      main.rundown.11pm     ;all & order off
```

Mailbox Tasks

For a server program to respond immediately to changes in a queue, you must set up mailbox tasks, by assigning the same mailbox number to the server and queue you want it to watch. Otherwise, the server will not know when changes are made to the queue.

As long as no stories are added to or changed in that queue, the server is inactive or “asleep.” However, when a story is modified in or added, the system wakes the server by notifying it through the mailbox. On notification, the server program checks its job list for mailbox tasks. For each mailbox task, it searches each scan queue to find modified stories and then executes the instructions in it and goes back to sleep.

A server places some additional load on your system when it wakes and executes a task. With mailbox tasks, you have no control over when the server wakes up. Use this kind of task only when the server must act as soon as a story is added to or modified in a particular queue.

In cases where the action can wait until a time when system use is light, use timed-interval tasks as explained in [“Timed-Interval Tasks” on page 329](#).

All mailbox tasks must precede any timed-interval tasks in a job list. If a queue identified in a timed-interval task has the server’s mailbox assigned to it, the server is awakened when changes are made to the queue but the queue is not scanned. That queue is scanned only at the designated time. The server scans all mailbox task lists whenever it wakes up.

Types of Mailboxes

The server program and the queue it watches must share the same mailbox. There are two kinds of mailboxes: reserved and standard.

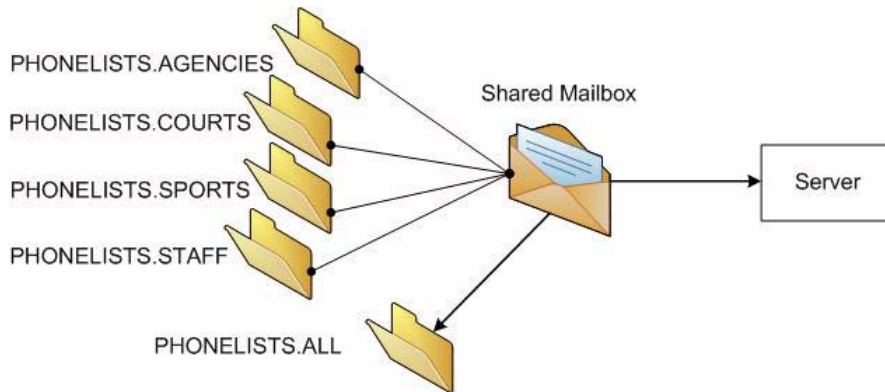
- Reserved mailboxes are used with special server programs reserved for system functions. See [“Reserved Mailboxes” on page 327](#) for more information.
- Standard mailboxes are used by server programs you can configure, such as action servers. There are 5000 standard mailboxes available, numbered 1 through 5000. When you create a server, choose a mailbox number that is the same as the server’s device number.



These mailboxes are not related to your system’s electronic mail feature, but are activation mechanisms or “signal carriers” by which servers are notified to perform predefined tasks.

When the system detects that a story has been modified in or added to a queue, such as PHONELISTS.PEOPLE, it checks to see if a mailbox is assigned to the queue. If so, it notifies any utility program sharing that mailbox a change has occurred in the queue. That message causes the utility program (in this case, an action server) to perform the task you have assigned to it, such as duplicate information from PHONELISTS.PEOPLE to PHONELISTS.ALL.

Several queues can be linked to a single server just by sharing the same mailbox number as the server program.



The following is an example of how the mailbox tasks would appear in the job list for the action server.

```
scan    phonelists.agencies
        dup phonelists.all
scan    phonelists.courts
        dup phonelists.all
scan    phonelists.sports
        dup phonelists.all
scan    phonelists.staff
        dup phonelists.all
```

Mailboxes are not limited to action servers, but can apply to other utility programs, such as distribution, parallel, or keyword servers. For instance, the WIRES.ALL queue can be assigned the same mailbox number as a distribution server, or you could set up a WIRES.AP-SEARCH queue in Wires with the same mailbox number as the keyword server.

The mailbox is an activation mechanism for a server program, so if a queue has a mailbox number matching a server program, then that server is the one activated or “awakened” whenever something happens to the queue.

Reserved Mailboxes

Reserved mailboxes are used only with servers reserved for your system to use. Each of these mailboxes links a specific system queue or directory to the server that watches that queue. Each queue that needs a reserved mailbox was assigned the correct one when your system was installed. The information presented here is intended to help you understand

how your system uses these reserved mailboxes. Unlike standard mailboxes, reserved mailboxes are referred to by name, not by number. The name and purpose of each of the reserved mailboxes appears in the following table.

| Mailbox | Purpose |
|----------------|---|
| All | Assigned to a queue used by your system to notify all users when an event occurs. The All mailbox should be assigned to the queue where urgent wire stories are sent. |
| Distribution | Assigned to the queue containing your system's distribution story for wires (usually SYSTEM.WIRES.DISTRIBUTION). This mailbox is used by the server program, known as the distribution checker, which distributes incoming wire stories to the appropriate destinations. |
| Group | Assigned to the system's group queue (SYSTEM.GROUPS). This mailbox is used by the server program, known as the group checker, which checks your system's group description stories for errors. |
| Keyboard | Assigned to the directory that contains your system's keyboard stories (usually SYSTEM.KEYBOARDS). It's assigned to the directory so that any queues created under it inherit the trait. This mailbox is used by the server program, known as keyboard checker, which checks your system's keyboard description stories for errors. |
| Keyword | Assigned to the queue with your system's keyword stories (usually SYSTEM.WIRES.KEYWORDS). This mailbox is used by the server program, known as the keyword checker, which checks your system's keyword stories for errors. |

Assigning a Mailbox to a Queue

You must assign a standard mailbox for a server to associated queues in order to set up that particular server.

To assign a mailbox to a queue:

1. Choose an available mailbox number (1 through 5000). This number should match the server's mailbox number. Also you must ensure no other device is using the chosen mailbox number by using the list c command at the console.

For instance, type: `list mailbox=2001 c`

Information similar to the following appears:

```
DEV DEVICE_TYPE COMPUTER NOTIFY OPTIONS DEVNAME
```


If you see the device configuration header with no information below it (as shown), then no device has that mailbox, and you can use that number. However, if configuration information for a device appears below the header, that device has the same mailbox as the one you chose. Therefore, choose another mailbox number.



Multiple queues can share the same mailbox number. To list all queues and directories in the database with a certain mailbox, use this format of the list d command:

```
list mail=<mailbox number> d
```

2. Log in to iNEWS Workstation as a system administrator or use the database manager login.
3. Assign the mailbox database trait to the queue.

Mailboxes are assigned to queues in the same way other database traits are—using the Queue Properties dialog box. See [“Changing Database Traits” on page 136](#), [“Database Traits Summary” on page 122](#), and [“Maintain Tab” on page 130](#) for more information.

Timed-Interval Tasks

Timed-interval tasks allow you to specify what time you want an action to take place. For instance, you may want an action server to watch a queue but not do anything until a certain time. Alternately, you may want an action server to monitor a queue and wait until everyone is done using that queue before performing some action. Timed-interval tasks are also used with rundown mirroring. See [“Configuring Rundown Mirroring” on page 340](#) for more information.

In addition to action servers, timed-interval tasks are also associated with tx links. However, this type of task is not used by other server programs, like distribution and parallel servers. Unlike a mailbox task, a timed-interval task acts like an alarm clock and wakes the server at a time you specify. On waking, the server examines the queues associated with that timed-interval task. If there are any new or modified stories in these queues, the server executes the commands in the tasks.




All mailbox tasks must precede any timed-interval tasks in a job list. If a queue identified in a timed-interval task has the server’s mailbox assigned to it, the server is awakened when changes are made to the queue but the queue is not scanned. That queue is scanned only at the designated time. The server scans all mailbox task lists whenever it wakes up.

For instance, you may create a timed-interval task that wakes the server at 3 a.m. to look for scripts added to or modified in the SCRIPTS.TODAY queue in the last 24 hours. If the server finds any, you could instruct it to move them to SCRIPTS.HOLDING.

When you create a timed-interval task, you must begin it with the time interval in which you want it to execute by using one or more of the following job list commands:

| Job List Command | Description |
|--|--|
| <code>at <hh:mm></code> | Sets the time of day when you want the task to wake up the server. Set the time in hours and minutes using a 24-hour clock where 12 a.m. (midnight) is 00:00 and 12 p.m. (noon) is 12:00. For instance, <code>at 3:30</code> wakes the server at 3:30 in the morning. If the time is after midnight but before noon, you do not have to put a 0 in front of the hour. |
| <code>every <D.HH></code> | Specifies an interval (the number of days and hours in D.HH format) that you want the task to follow when waking the server. For instance, <code>every 0.12</code> wakes the server every 12 hours. |
| <code>on <day> [<day>...]</code> | Names the day or days on which you want the task to wake up the server. For instance, <code>on mon tue</code> wakes the server on Monday and Tuesday only. |

 *Names of the days of the week are defined in `/site/dict/words`.*

You can combine these commands to create very specific time intervals. For instance, a timed-interval task created to execute at 10 a.m. every weekday would begin with the following time interval:

```
at 10:00

on mon tue wed thu fri
```

Example of Timed Interval Tasks

Here is an example of how an action server's job list uses timed-interval tasks to move old scripts out of each show's script queue after the show has aired:

- Separate timed-interval tasks for each show wakes the server 15 minutes after the show has aired. Because the three shows are not aired on weekends, the timed-interval tasks wake the server only on weekdays.
- The morning show ends at 8 a.m., so the first timed-interval task could wake the server at 8:15 a.m. and move the scripts in `SHOW.MORNING.SCRIPTS` to `SCRIPTS.TEMP-ARCHIVE`.
- The noon show ends at 1 p.m., so the second timed-interval task could move the scripts in `SHOW.NOON.SCRIPTS` to `SCRIPTS.TEMP-ARCHIVE` at 1:15 p.m.
- The evening show ends at 6 p.m., so the third timed-interval task could move the scripts in `SHOW.EVENING.SCRIPTS` to `SCRIPTS.TEMP-ARCHIVE` at 6:15 p.m.

Each timed-interval task must begin with a time interval indicating when you want it to wake the server. This is followed by a scan line identifying the queue you want the server to examine on that day and time. Finally, the scan line is followed by one or more tasks to be performed on stories in that queue at that time.

You can have multiple command sets controlled by a single timed-interval. In this example, the finished tasks look like this:

```
at 8:15
on mon tue wed thu fri
    scan show.morning.scripts
    move scripts.temp-archive
at 13:15
on mon tue wed thu fri
    scan show.noon.scripts
    move scripts.temp-archive
at 18:15
on mon tue wed thu fri
    scan show.evening.scripts
    move scripts.temp-archive
```

Action Servers

An action server is a general-purpose utility program that automatically performs specified routine tasks, such as moving or copying stories added to or edited in certain queues, according to a job list you define. One action server can perform multiple tasks not necessarily related to each other. Here are some examples:

- Example: You want an action server to update a phone list. You have organized several queues as phone lists: PHONELISTS.AGENCIES, PHONELISTS.COURTS, PHONELISTS.STAFF, PHONELISTS.SPORTS, and PHONELISTS.PEOPLE. Additionally, you have created another queue as a master phone list, PHONES.ALL, that holds all numbers in the individual source phone queues. What you want the action server to do is have PHONES.ALL updated whenever anyone adds or changes a number in one of the source phone queues.
- Example: You also want the action server to make a daily archive of all the scripts used in your morning, noon, and evening news shows. After each show, you want to move scripts in that show's script queue to the queue SCRIPTS.ARCHIVE-TEMP. At the end of the day, you want to move everything in this queue to an archive queue named SCRIPTS.MAY.01 on May 1, SCRIPTS.MAY.02 on May 2, and so forth. You want the action server to automatically perform all of these activities for you, rather than having to do them manually.
- Example: You want to replicate data from a primary queue to a backup queue, including any story additions, deletions, modifications, and reordering, on the same system or through a tx link to another system's iNEWS Server or to an external third-party FTP server.

In the first bulleted example, mailbox tasks are required to achieve the desired results. The server shares the same mailbox of the queue(s). See a visual representation of how an action server uses mailbox tasks to maintain a master phone list in [“Mailbox Tasks” on page 325](#).

In the second example, timed-interval tasks are needed. See [“Example of Timed Interval Tasks” on page 330](#) for another example of an action server using timed-interval tasks to archive scripts. A timed-interval action server operates on a queue at specified times without requiring a shared mailbox.

In the final example, polling is required for the rundown mirroring process. See [“Rundown Mirroring” on page 339](#) for more information.

Adding an Action Server

Action servers can perform multiple or single tasks. They can be time-related or respond to a system change. In this section, we will outline a procedure where we will create an action server that performs the following activities:

- Update the master phone list as soon as anyone makes a change to any of the source phone lists.
- Move each show’s scripts to SCRIPTS.ARCHIVE-TEMP at the end of the show.

To add an action server to your system:

1. Choose a device number and mailbox number for the server which will also be used for associated queues. In this procedure, 256 is used as the example.
2. Add the server (utility program) to the configuration file on each iNEWS Server—such as servers A and B—in your system.

For instance: `server 256 action 256 - ;action server`

Details for this step are provided in [“Adding a Server Program to the System” on page 316](#).

3. Create a job list queue in the Configure folder of the System directory in the iNEWS database. The queue’s name, such as 256, is the number selected earlier.

For instance, the full pathname is SYSTEM.CONFIGURE.256.



Since the action server has a number of tasks, you can end the name with a hyphen and a descriptive word, such as “general.” The pathname would then look like: SYSTEM.CONFIGURE.256-GENERAL. See [“Creating a New Queue” on page 104](#) for more information.

4. Create a job list story with tasks for the server.
 - a. Open the job list queue, such as SYSTEM.CONFIGURE.256.
 - b. Create a new story, the first in the job list queue, to hold the action server’s job list. See [“Creating a New Story” on page 111](#) for more information.
 - c. Add tasks you want the server to perform; mailbox tasks first, followed by timed-interval tasks. Ensure each task in the job list story has a scan line identifying a queue to scan, followed by job list commands the server should execute. See [“Job Lists: Queues, Stories, and Commands” on page 320](#) and [“Job List Commands” on page 527](#) for more information on tasks and job list commands.

Here is a job list story with mailbox tasks:

```

title          presenter
General Server
scan  phonelists.agencies
      dup phonelests.all
scan  phonelists.courts
      dup phonelists.all
scan  phonelists.people
      dup phonelists.all
scan  phonelists.sports
      dup phonelists.all
scan  phonelists.staff
      dup phonelists.all

```

Here is a job list story with timed-interval tasks added:

```

title          presenter  writer
General Server          user5000
dup phonelists.all
At 8:15
On Mon tue wed thu fri
  scan show.morning.scripts
  move scripts.temp-archive
At 13:15
On Mon tue wed thu fri
  scan show.noon.scripts
  move scripts.temp-archive
At 18:15
On Mon tue wed thu fri
  scan show.evening.scripts
  move scripts.temp-archive

```



You can have one server watch many different queues by putting a separate task for each queue in the server's job list. You can have one server perform several tasks or have separate servers perform each task, depending on the task. The number of jobs—a job is a scan plus the actions for that scan—for a mailbox activated server should be as small as possible because a change in a single queue could cause the server to do a lot of work; when activated it will scan all queues in the job list, excluding timed actions.

Ideally each queue should have a server, but because of system resource limitations, this organization is rarely possible. Group queues according to the rate they are generally updated (rarely, moderately, or frequently) and assign servers to the groups.

5. Assign mailbox database traits to the appropriate queues, using the number matching the mailbox assigned earlier to the server in the configuration file.
 - a. Open the Queue Properties dialog box for a scan queue, such as PHONELIST.AGENCIES. See [“Changing Database Traits” on page 136](#) for more information.
 - b. Click on the Maintain tab. See [“Maintain Tab” on page 130](#) for more information.
 - c. Select the Standard radio button.
 - d. Type in the mailbox number, such as 256.
 - e. Click OK to save changes.
6. Test your configuration changes. See [“Testing the Site Configuration File After Alteration” on page 253](#) for more information.
7. Reconfigure the system. See [“Incorporating Configuration Changes” on page 253](#) for more information. When adding server programs, you do not need to stop anything to reconfigure the system.
8. Start the server program using the following command format: `restart <device #>`
For instance: `NRCS-A$ restart 256`
9. (Optional) Back up site files.

Field Validation

Validation is an option for use with action servers and tx links. With it, you can provide an action server or tx link with a list of acceptable values for certain fields. When the action server or tx link processes a story, it checks the contents of these fields to make sure each contains an entry you have defined as acceptable for that field. See [“Adding Rx/Tx Links” on page 410](#) for more information.

If all fields you want checked contain acceptable entries, the server or tx link continues to process the story according to the command set for the task it is executing. However, if one of the fields you want checked contains an invalid entry, the action server or tx link puts the story in an error queue and ignores the remaining commands in the task.

Possible Uses of Validation

Validation can help users attach accurate distribution codes to stories before sending them to a distribution server. See [“Distribution Servers” on page 344](#) for more information. Part of its function is to combine entries in each field it checks into a distribution code. If all fields contain acceptable values, it attaches that code to the story. You can have the action server or tx link use a move or dup command to overwrite this code with another one.

See [“Distribution Codes” on page 345](#) for more information.

This way, a user can put a code the user wants to attach to a story in one of the story's fields. The user can send the story to an action server or tx link that is set up to check that field. If the field contains a valid distribution code, the action server or tx link attaches that code to the story and sends the story to a distribution server. Otherwise, it sends the story to an error queue and notifies the user of the error.

You can also use validation to allow people to enter different parts of a distribution code in different fields. If the action server or tx link determines that these fields contain acceptable values, it combines the contents of the fields into a distribution code, which it attaches to the story before sending it to a distribution server.

Though validation was designed primarily to help create accurate distribution codes, you can use it anywhere you need to have an action server or tx link verify the contents of one or more of a story's fields before processing the story. For instance, you could use validation to have an action server check the created by and modified by fields in each story it processes to ensure that it accepts only stories created or modified by certain users.

It could also be used to rid the rundown of unwanted pages before archiving by searching for a specific string in a field and removing any stories with that string. Any stories the producer did not want or need to archive could be coded with that string and later stripped from the rundown by a timed-interval action server using validation.

Using Field Validation

You can use validation with an action server or a tx link. Adding validation to either uses the same process.

For example, set up the tx link to send stories marked with a distribution code indicating whether they are news or sports stories and whether they are intended for television or radio. First, create a form writers can use to mark stories as either news or sports stories and indicate whether they are intended for television or radio.

The writer indicates whether the story is for television or radio by selecting from a drop-down list or typing either tv or radio in the TV/RADIO field. Likewise, the writer indicates whether the story is a news or sports story by typing either news or sports in the NEWS/SPORTS field.

Set up the tx link to check each of these fields. If it determines both contain valid entries, it combines the contents of these fields into a distribution code. Then it attaches that code to the story and sends the story to the affiliate.

To use the validation feature with a server:

- ▶ Create a validation story containing instructions you want the server to use for validation. This story can be in any queue, such as `SYSTEM.VALIDATION`.

The validation story consists of one or more verification instructions. Each verification instruction identifies a field you want the server to check and specifies a list of values that field is allowed to contain. The format for a verification instruction is:

```
verify <field name> <length> [ignore]
<list of allowed values for the field>
```

- a. Begin each verification instruction with the word **verify** followed by the name of the field you want to check. For instance, put **title** here to check a TITLE field.
- b. Specify the number of characters you want to check in that field. For instance, to check three characters, type **3**. The server compares only the first three characters that appear in that field against the list of acceptable values for the field.

This number also controls how many characters in a field are used in the distribution code. If you specify three, only the first three characters in the field are used. The server converts any space characters (trailing or embedded) to hyphens so that it can add them to the distribution code.

- c. (Optional) Use the ignore validation story in an instruction to indicate whether—if the field is validated—the server should skip validation of the other fields and build the distribution code as if all fields contained acceptable values. If a preceding field failed validation, the story will be considered invalid.

If the field fails verification, the server moves to the next field to be verified and does not consider the story to be invalid. You can use ignore only once in a validation story.

- d. List the values you allow for the field. The value list must start on the line following the verify command.

If you list more than one value, separate them with spaces. List as many values on as many lines as necessary. Include plus and minus characters to indicate ways in which you want the field to be checked.

- A plus sign (+) means the field can be blank.
- A minus sign (-) means the field can be any value except blank.
- A plus and minus sign (+ -) used together means that any value is acceptable for the field. This is not the same as using the ignore option on that field.



For clarity, provide more characters for each item in the list of acceptable values for a certain form than you have specified in the verify line. The server ignores the extra characters. You can list possible values along with the + - option, which tells the server to accept anything in that field as valid. Any values you list in addition are superfluous, but harmless.

Validation Job List Commands

An action server or tx link has a job list containing a separate task for each queue you want it to watch. If you want an action server or a tx link to validate fields, you must put the appropriate validation commands in each task of the job list where you want it to check the fields.

The validate command must appear after the task's scan command. Any commands preceding the validate command are processed before the validate command; therefore, those commands are not dependent on the outcome of the validation. Two commands, ignore and quiet, alter the behavior of the validate command. If present, they must precede the validate command to affect it.

```
quiet no
```


```
ignore yes
```

```
validate <validation queue name> <error queue name>
```

These job list commands are described as follows:


| Job List Commands | Description |
|-------------------|--|
| quiet no | An action server or tx link that validates fields displays a message only if it encounters a field that does not contain an acceptable value. Otherwise, it processes the story. If you include the quiet no validation instruction in the task, the action server or tx link also displays a message indicating success whenever it successfully validates a story. |
| ignore yes | Occasionally, you may want to create a task that builds distribution codes from the contents of certain fields, but does not validate those fields. Include the ignore yes validation instruction in the task. Then, the action server or tx link accepts whatever appears in the fields you have it check. |

| Job List Commands (Continued) | Description |
|--|--|
| <code>validate <validation queue name> <error queue name></code> | <p>Include this instruction in the task for the action server or tx link to validate fields when it executes the task. This instruction must begin with the word <code>validate</code> followed by the queue containing the validation story that you want the server to use with that task. Then, list the error queue where you want the action server or tx link to place stories that it cannot validate.</p> <p>If you use either <code>ignore yes</code> or <code>quiet no</code>, they must precede this instruction.</p> |

 *The <error queue name> is subject to strftime formatting.*


Rundown Mirroring

Rundown mirroring takes what's written in a primary queue and copies it to a backup queue like a one-way mirror. All production activities—conducted only in the primary queue—including story additions, deletions, modifications, and reordering are replicated to the backup queue by a timed-interval action server configured for polling. Rundown mirroring is also available for tx links.

 *Conducting production activities directly into the backup queue is not supported and may cause undesirable behavior. Avid suggests assigning a restricted write group, such as “nobody-writes” to all designated backup queues. This write group restriction will not prevent the timed action server from completing its polling and replicating tasks.*

Here are some additional guidelines for mirroring rundowns:

| Feature | Primary Rundown | Backup Rundown |
|-----------------------|-----------------------|------------------------|
| Story and queue forms | Same as backup | Same as primary |
| Mailbox number | Different than backup | Different than primary |

 *Rundown mirroring allows for two monitor servers to load data from identical rundowns to separate iNEWS Command Servers, possibly controlling separate video servers. For this to work, each rundown is linked to a separate monitor servers through different mailbox numbers.*

| Feature (Continued) | Primary Rundown | Backup Rundown |
|---------------------|---|--|
| Purge interval | May have a zero purge interval to allow producers to manually delete the previous day's contents. | Must have a purge interval that ensures the previous day's contents are purged prior to production and rundown mirroring the next day. |
| Update trait | Not enabled | Must have the update trait enabled. |

Polling Commands for Action Servers/Tx Links

Two job list commands, poll and bpoll, allow iNEWS to poll a local source queue for a specified duration of time at specified intervals. The source queue is scanned at every interval and stories with modified times greater than the time of the last scan are processed. Polling stops when the system time is greater than the time of the initial poll plus the polling duration. The poll command scans the queue in a forward direction, while bpoll scans the queue in a reverse direction. The syntax for both commands is as follows:

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

If the polling interval and/or polling duration are omitted, a single scan of the queue will be done. The syntax for the polling interval and polling duration is hh:mm:ss, mm:ss, or ss. For the hh value, the range is 0 to 23. For the values of mm and ss, the range is 0 to 59. Both commands are restricted to a timed action set. They are not available as mailbox triggered actions.

Configuring Rundown Mirroring

The configuration procedure for rundown mirroring involves steps conducted at the console and at an iNEWS Workstation.

To configure rundown mirroring:

1. Determine what action servers are configured on your system and are available to be used for rundown mirroring. At the console, you can use the `grep` command to pull out all the lines from the `/site/config` file that have the word "action" in them:

```
NRCS-A# grep action /site/config
server 244 action 244 - ;
server 245 action 245 - ;
server 251 action - - ; timed-action
server 252 action - - ; timed-action
server 253 action - - ; timed-action
```

In the example above, five action servers are outlined in the configuration file. Those numbered 251 - 253 have no mailbox assigned to them, having a dash in the fourth column. Normally action servers are linked via a shared mailbox number to a queue. But that is not necessary for timed-interval action servers. For the purposes of this procedure, 252 will be used.



An alternative to the `grep` command is to type: `list c action`.

2. Create a job list queue for the timed-action server in the `SYSTEM.CONFIGURE` directory. Its name must start with the number of the server. You can add a hyphen, then a descriptive name. For instance, any of the following would be valid queue names for an action server numbered 252.

```
SYSTEM.CONFIGURE.252-ACTION
SYSTEM.CONFIGURE.252-MIRRORING
SYSTEM.CONFIGURE.252-10PM-NIGHTCAST-MIRROR
SYSTEM.CONFIGURE.252-TIMED-ACTION
```

Action server 252 will look for its job list in the queue you create. The job list is a listing of commands the action server will perform. See “Creating a New Queue” on page 103 for more information.

3. Create a new story in the queue, if one does not exist, and start writing the job list. Here is a sample job list for action server 252:

```
at 20:30
poll SHOW.10PM-NIGHTCAST.RUNDOWN 00:00:05 3:30:00
order yes
dup SHOW-MIRROR.10PM-NIGHTCAST.RUNDOWN
```



Timed-interval action servers work off the iNEWS Server's clock. All times must be listed in 24 hour time. So, 20:30 in the above example instructs the action server to begin at 8:30 PM.

The sample action server is instructed to poll the Nightcast rundown queue once every 5 seconds (00:00:05) for a total of three and a half hours (3:30:00) before automatically turning off at midnight (20:30 start time + 3:30:00 duration).

You may use the `bpoll` command instead of `poll` to instruct the action server to begin the scan at the opposite end of the queue, in which case the second command line of the job list would appear as follows:

```
bpoll SHOW.10PM-NIGHTCAST.RUNDOWN 00:00:05 3:30:00
```

The `order yes` command ensures that re-ordering events in the polled queue get transmitted to the backup queue. Without this, the timed-interval action server would not pick up any reordering of the rundown by the producer. The primary rundown queue and the backup rundown queue would have the same stories, but they would not be in the same order after the sequence of stories was changed in the primary rundown. See [“Ordered Queues and the Order Command” on page 325](#) for more information.



The order yes command is only relevant when polling to or from an iNEWS Server. It is not possible to "order" stories mirrored through a tx link to an external, third party FTP server.

Overlapping Job Lists

Each joblist in a SYSTEM.CONFIGURE queue can have more than one polling instruction servicing more than one rundown. For example, the job list for timed-interval action server 252 might look like this:

```
at 09:30
poll SHOW.NOON.RUNDOWN 00:00:05 3:30:00
order yes
dup SHOW-MIRROR.NOON.RUNDOWN
;
at 15:30
poll SHOW.6PM.RUNDOWN 00:00:05 3:30:00
order yes
dup SHOW-MIRROR.6PM.RUNDOWN
;
at 20:30
poll SHOW.10PM-NIGHTCAST.RUNDOWN 00:00:05 3:30:00
order yes
dup SHOW.10PM-NIGHTCAST.RUNDOWN
```

The only restriction is that none of the polling actions in a job list for a single timed-interval action server can overlap. Since the Noon show begins at 09:30 and runs for three and a half hours, polling won't be done until 1 PM (or 13:00). So another show cannot be scheduled to start in this job list until after 1 PM.

If an overlap in polling is necessary, separate job lists for separate timed-interval action servers are required.

Polling Issues Related to Tx Links

Rundown mirroring may be used to poll rundown queues in an iNEWS system and replicate the data to another system's iNEWS Server or to a third-party FTP server. The guidelines for mirroring still apply, such as overlapping job list restrictions, purge intervals, and update traits. The following are examples of job lists for rundown mirroring through a tx link:

```
at 15:55
poll SHOW.4PM.RUNDOWN 00:00:05 1:00:00
open myftpserver user3 html system.webforms.s.scripts
put /shows/4pm
```

The above example would begin polling the SHOW.4PM.RUNDOWN at 3:55 PM, at five-minute intervals for one hour, then open a connection to an external FTP server using the name, user3, and place stories in the /shows/4pm folder in HTML format based on the HTML skeleton outlined in the SYSTEM.WEBFORMS.S.SCRIPTS queue. See [“Web Publishing” on page 430](#) for more information.



FTP servers do not order stories, so the `order yes` command line is not valid. Also, the username in the job list (user3, in the above example) must exist as an account in the iNEWS database and on the external FTP server. It must have the same password on both systems, and have proper permissions on the FTP server to create and overwrite files. If the user account is only used for outbound tx links, it may be blacklisted on the iNEWS Server to prevent it being used for actual logins.

```
at 15:55
poll SHOW.4PM.RUNDOWN 00:00:05 2:00:00
order yes
open AVID-A transmituser nsml
put SHOWS.MADISON.4PM.RUNDOWN
```



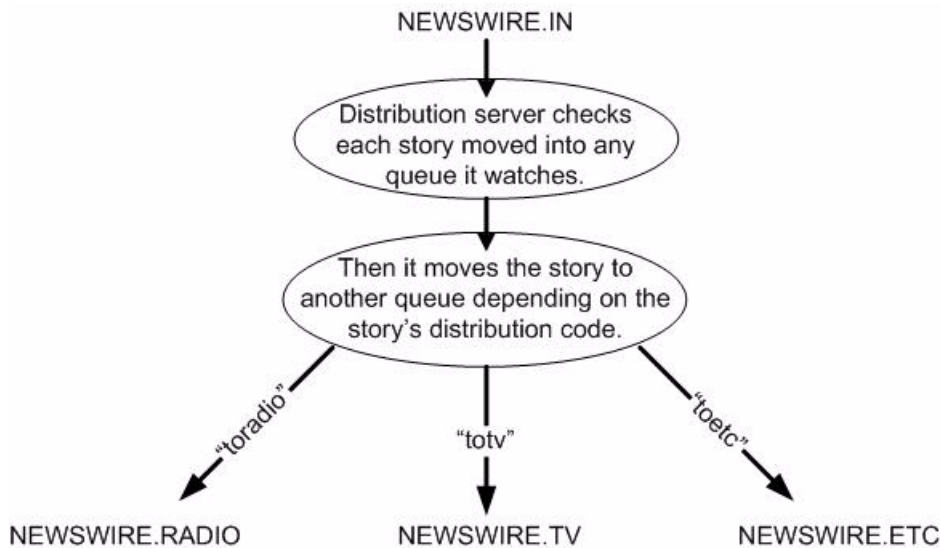
News Story Markup Language (NSML) is the default format for transmitting to iNEWS Servers, so the `nsml` on the open command line is optional. Also, the pathname for the put command line uses periods (.) when transmitting between iNEWS systems, unlike the slashes (/) used when transferring to an FTP server.

Distribution Servers

A distribution server is used to route stories based on their distribution codes. Distribution codes are text strings of up to 32 characters which you can assign to your stories. A distribution code longer than 32 characters is truncated to 32 characters before being assigned to the story. When you assign a distribution code to a story, you overwrite any distribution code that story may already have.

Distribution servers are usually used by large stations or networks who write their own wire stories and send them to their affiliates. By attaching distribution codes to these stories, their affiliates can distribute the stories to appropriate queues.

Distribution servers can also monitor a queue. When a new story is added to a queue, the distribution server will move the story to another queue based on the distribution code and instructions you assign.



Also, distribution instructions in a distribution story can use the same always, silent, urgent, flash, and bulletin options that wire service distribution instructions use. These options must appear at the end of the distribution instruction if you use them.



A distribution server is different from an action server in that it will always remove each story from the source queue after distributing it. An action server depends on the move and remove commands.

Distribution Codes

You can create a distribution instruction that matches more than one distribution code by using a wildcard, which is a special character—a pound sign (#) in this case—that matches any character.

For instance, the following distribution instruction tells the distribution server to match all code that begin with “tv” and send the story to INEWSWIRE.TVNEWS:

```
tv#          inewswire.tvnews
```

When the wildcard character is placed at the end of the pattern, the distribution server interprets it as an instruction to match all remaining characters. If it occurs anywhere else in the pattern, the distribution server matches only characters that occur in that position.

- `tv#` – matches anything that begins with tv
- `AP#a` – matches any code that begins with AP and contains the letter a in the fourth position.

When the distribution server is trying to match a story’s code to an instruction line, it stops at the first matching line. It then looks at the next line to see if it matches as well. If it does, it looks at the next line. Once it sees a line which does not match, it stops looking and distributes the wire to the destinations named on the matching lines.

Suppose you have the following two instructions, and the distribution server is trying to match a story with distribution code tvsports.

```
tv#          inewswire.all
tvsports     inewswire.tvsports
```

The distribution server matches the first instruction that contains the pattern `tv#`, and then it looks at the next line to see if it also contains the pattern `tv#`. In this case, there is no such line, so it sends the story to INEWSWIRE.ALL and not to INEWSWIRE.TVSPORTS.

If you use just a wildcard character as a distribution code, that instruction should come at the end of the job list story. You would do this to create an instruction for all stories with distribution codes that do not match any other distribution instructions. After the server matches a wildcard only, it stops looking for additional matches for the story.

Wildcards and the Destination Queue

You can also use a wildcard as a part of a distribution instruction’s destination queue. Whenever the distribution server uses that instruction line, it replaces the wildcard with the distribution code attached to the story it is processing.

For instance, suppose that your distribution story has the following instruction line:

```
tvnews    inewswire.#
```

In this case, whenever it processes a story with the distribution code tvnews, it uses this line. In doing so, it uses the story's distribution code in place of the wildcard in the destination queue and distributes stories with the distribution code tvnews to INEWSWIRE.TVNEWS.

This is most useful when you also use a wildcard in the instruction's distribution code. You can create an instruction that handles stories with several different distribution codes.

For instance, suppose you want your distribution server to send stories with the distribution code tvnews to INEWSWIRE.TVNEWS and stories with the distribution code tvsports to INEWSWIRE.TVSPORTS. You can do this with a single instruction if you use a wildcard in the instruction's distribution code and destination queue:

```
tv#       inewswire.#
```

This instruction causes the server to copy each story with the distribution code tvnews to INEWSWIRE.TVNEWS. Likewise, it causes the distribution server to copy each story with the distribution code tvsports to INEWSWIRE.TVSPORTS.



If you try this, ensure that what gets substituted into the wildcard portion of the destination queue forms the pathname of an existing queue. Otherwise, the server cannot copy stories to that queue and sends them to SYSTEM.UNKNOWN, or the default error queue you named. If the queue does not exist, it will be created. If any directory in the path does not exist, it will not be created, and the stories will be placed in SYSTEM. UNKNOWN.

Move and Dup Commands

When moving or dup'ing stories from a session on an iNEWS Workstation, you can use the move and dup commands to attach a distribution code to a story. If people on your system use these commands frequently to attach distribution codes to stories, publish the codes you want them to use.

Action Servers or Tx Links

You can attach distribution codes to stories that you send to a distribution server using an action server or a tx link. There are two ways to use action servers and tx links to do this.

- Dup or Move commands
- Validate commands



Tx links are explained in "Network iNEWS Systems Using RX/TX Links" on page 407.

Dup or Move Commands in Job Lists

You can use the move and dup commands in action server and tx link job lists to attach distribution codes to the stories those servers process. For instance, the following job list shows how you may have an action server attach a distribution code to any story added to TVNEWS.OUT:

```
scan tvnews.out
    dup inewswire.send tvnews
```

The job list tells the server to attach the code tvnews to each story it handles and then copy the story to INEWSWIRE.SEND.

Use this method to automatically assign a particular distribution code to every story added to a particular queue. If you need more flexibility in how the server assigns codes, see [“Distribution Codes” on page 345](#).

Validate Commands in Job Lists

Action servers and tx links support a feature called validation. This feature allows you to have an action server or tx link extract a distribution code from a story’s form and check whether it is a valid code. If the code is valid, the server attaches it to the story and sends the story to the distribution server. If the code is invalid, the server puts the story in an error queue and notifies the user. Validation is covered in more detail in [“Field Validation” on page 335](#).

Instructions in the Wire Distribution Story

A wire program relies on instructions in the wire distribution story to determine what to do with each story it receives. This allows the wire program to use each story’s wire service code to determine where to send the story.

Each wire program can have no more than 150 distribution instructions. If you need to handle more than 150 codes from a particular wire service, you can use a distribution server to get around this limit.

Add special distribution instructions to the wire distribution story. These instructions cause the wire program to look for stories with the wire service codes you want the distribution server to handle. The wire program assigns each story a distribution code matching its wire service code and sends it to the distribution server’s source queue.

Those instructions should match the following format. They should begin with a wire service code that you want the distribution server to handle. Follow the code with the distribution server's source queue. You must follow the destination queue with the TRANSMIT option:

```
<wire service code> <destination queue name> TRANSMIT
```

When a story comes in with that wire service code, the wire program gives the story a distribution code that matches its wire service code. Then the wire program sends the story to the destination queue specified in the instruction.

You can use options such as ALWAYS, SILENT, and URGENT with the TRANSMIT option. A distribution instruction that assigns stories distribution codes matching their wire service codes does not have to send those stories directly to a distribution server. The instruction can send them to an intermediate queue and then an action server can send them to the distribution server.

Matching and Case

When using two or more distribution instructions that begin with the same code, you must enter the codes in the same case. If you do not, the distribution server does not execute all the instructions when it receives a story with that distribution code.

When a distribution server processes a story, it begins by checking whether the first instruction in its distribution story matches the story's distribution code. If not, the server checks the second instruction, and then the third, and so on, until it finds one that matches the story's distribution code.

When it finds a match, it sends the story to the queue specified in that instruction. Then the distribution server does a case-sensitive search on the remaining instructions in the distribution story to see whether any use exactly the same distribution code (that is, the same code entered in the same case) as the instruction that it has matched. If any instructions match, the server sends the story to the queues specified in those instructions as well.

For instance, if a distribution server successfully matches an instruction with a distribution code of tvnews, it looks for other instructions that also begin with tvnews and sends the story to the queues specified in those instructions. It ignores instructions that have that code in a different case, such as TvNews or TVnews.

Matching and Order

The order of instructions is important, especially when you have a distribution code that also forms the beginning of another distribution code, such as tv and tvnews. In this case, you need to make sure that instructions that use the longer code appear before those that use the shorter code.

When the distribution server attempts to match an instruction's code to a story's distribution code, it considers that it has a match if the instruction's code matches some initial portion of the story's distribution code. For instance, the distribution server would match an instruction for code `tv` to a story with distribution code `tvnews`.

After the distribution server has matched an instruction code to the story, it stops trying to match additional instructions to the story. Instead, it looks for other instructions that have the same code as the instruction code that it has matched. It never matches the instructions with the longer codes.

If you had a distribution story with the following instructions in the order shown below, and if you were to send a story with distribution code `tvnews` to this distribution server, the distribution server would match the first instruction in the distribution story but would never execute the second:

```
tv      newswire.tv
tvnews  newswire.tvnews
```

In this example, the server always matches the first instruction to any story that has a distribution code that begins with `tv`. Then it looks for other instructions that have `tv` as their code. It never executes the instruction that has the code `tvnews`. Always arrange such instructions so that those with longer codes come before those with shorter codes. That way, only stories with the distribution code `tvnews` (or longer) match the first line. Stories with the code `tv` fall through the first line and match the instruction in the second line.

Adding a Distribution Server

Additional details for the first two steps in the following procedure are provided in [“Adding a Server Program to the System” on page 316](#).

To set up a distribution server to watch a queue and distribute stories placed in a queue:

1. Choose a device and mailbox number for the server that will also be used for associated queues.
2. Add the server (utility program) to the configuration file on each iNEWS Server—such as servers A and B—in your system.

For instance: `server 257 distribution 257 - ;distribution server`

3. Create a job list queue in the Configure folder of the System directory in the iNEWS database. The queue’s name, such as 257, is the same as the device and mailbox number selected earlier. For instance, the queue’s full pathname is `SYSTEM.CONFIGURE.257`.

You can also end the name with a hyphen and a descriptive word, such as `distribution`, in which case the queue’s full pathname would then look like this:
`SYSTEM.CONFIGURE.257-DISTRIBUTION`.

4. Create a job list story for the server, by doing the following:
 - a. Open the job list queue, such as `SYSTEM.CONFIGURE.257`.
 - b. Create a new story, the first in the job list queue, to hold the distribution server’s job list. See [“Creating a New Story” on page 111](#) for more information.
5. Add tasks that you want the distribution server to perform; include a separate task for each queue the server watches.

Each task in the job list story consists of two lines with the following format:

```
source <queue scanned by server>
distribution <distribution queue> [<error queue>]
```

| Command | Description |
|---------------------------|--|
| <code>source</code> | Identifies which queue the distribution server should scan. This job list command is the same as the scan command, used by action servers. The queue is known as both the scan queue and source queue. |
| <code>distribution</code> | Specifies the queue containing the distribution story and, optionally, an error queue for stories whose distribution codes cannot be processed. |



If you do not specify an error queue, the distribution server sends any such stories to `SYSTEM.UNKNOWN`.

For instance, the job list story may appear like this:

```
source newswire.in
distribution system.distribution.257 newswire.error
```



Unlike action servers where the scan queue identifier (scan line), tasks, and commands (instructions) are located altogether in the job list story, a distribution server separates the instructions (except for the distribution command), placing them in another queue and story, called the distribution queue and distribution story, respectively. The distribution queue is typically given the same numerical name as its corresponding job list queue and should be located in a directory, typically called SYSTEM.DISTRIBUTION.

6. Create the distribution queue and story, by doing the following:
 - a. Navigate to SYSTEM.DISTRIBUTION. (If this directory does not exist, create it. See [“Creating a New Directory” on page 102](#) for more information.)
 - b. Create a new queue, giving it the same numerical name as the job list queue for the distribution server, such as 257. The full pathname for the new queue would be SYSTEM.DISTRIBUTION.257. See [“Creating a New Queue” on page 104](#) for more information.
 - c. Open the new queue and create the distribution story.

Each instruction in the distribution story is one line long and has the following format:

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

For instance, suppose we know that some stories sent to our distribution server have tvnews as their distribution code and others have tvsports. We want the distribution server to send stories with the tvnews code to INEWSWIRE.TVNEWS and those with tvsports to INEWSWIRE.TVSPORTS.

In this case, you would type the following two instructions in the distribution story:

```
tvnews inewswire.tvnews
tvsports inewswire.tvsports
```

If you want to send stories that have a certain distribution code to more than one queue, follow this example:

```
tvnews inewswire.news
tvnews inewswire.all
tvsports inewswire.sports
tvsports inewswire.all
```

The choices for the options parameter are: ALWAYS, BULLETIN, FLASH, SILENT, and URGENT.

7. Assign mailbox database traits, using the number matching the mailbox number chosen earlier in this procedure to the appropriate source queue(s)—the queue(s) to be scanned by the server.

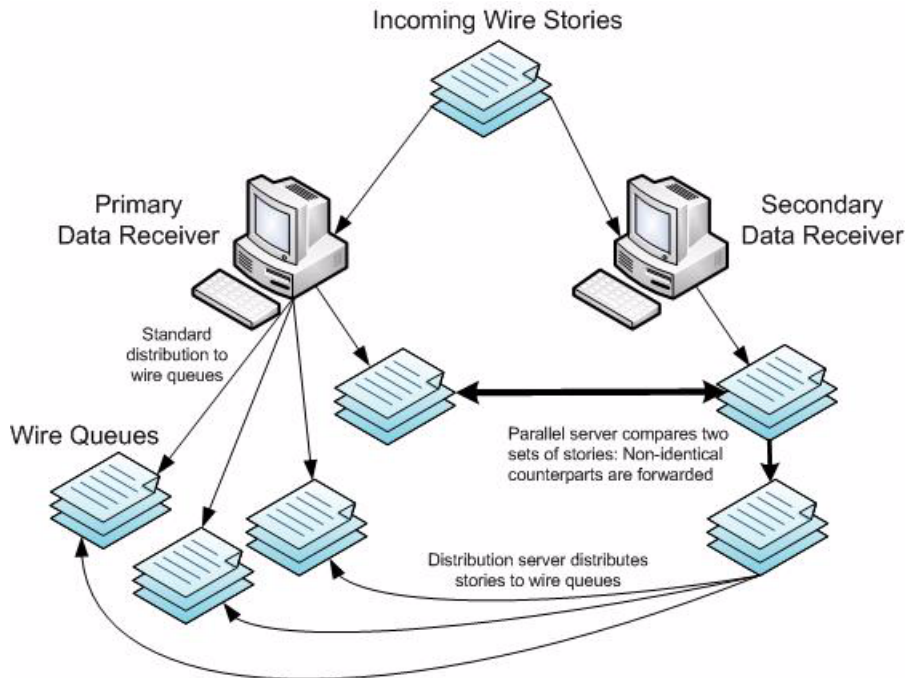
For instance, the job list queue is named 257, so that is the standard mailbox number assigned to queues identified in the job list associated with the distribution server.

- a. At an iNEWS Workstation, open the Queue Properties dialog box for a scan queue, such as NEWSWIRE.IN.
 - b. Click on the Maintain tab.
 - c. Select the Standard radio button.
 - d. Type in the mailbox number, such as 257.
 - e. Click OK to save changes.
8. Repeat the previous step as needed for each source queue in the job list.
 9. (Optional) Test your configuration changes. See [“Testing the Site Configuration File After Alteration” on page 253](#) for more information.
 10. Reconfigure the system.
 11. Start the distribution server.

When you see the `System being configured` message, select the iNEWS Server where the distribution server is configured and start the distribution server. For instance, to start a distribution server with device number 257, type: `restart 257`
 12. (Optional) Back up site files using `sitedump` command.

Parallel Wire Servers

Another use for a distribution server is to help provide backup wire distribution in case the usual wire distribution fails—if, for instance, the Data Receiver receiving the wire goes down. This is accomplished by a distribution server working in tandem with a parallel wire server.



A parallel wire server works as follows:

- The same wire service is received through two different wireservers, preferably from two different Data Receivers sending to two different iNEWS Servers. One of these wires uses the standard wire distribution feature, with a line added to distribute its wires

to the queue monitored by the parallel wire server. The second wire uses one line in the distribution story, which routes its wire stories to the queue monitored by the parallel wire server.

- The parallel wire server checks to make sure that each story received from the secondary wire has a counterpart from the primary wire. If a secondary wire story cannot be matched, it is passed on to a distribution server. No action is taken on unmatched primary wire stories, since they were already distributed normally.
- If the primary wire fails, each secondary story your system receives lacks a corresponding primary story, so the parallel wire server passes each one to a distribution server. This server uses a distribution story nearly identical to the wire distribution story, so stories are routed to the same queues as with standard wire distribution. Wire reception thus appears normal to users.

Although using two wireservers for the same wire may seem like a waste of a device, this redundancy can be a lifesaver in the long run. However, a parallel wire server is useful only for problems within the system such as a computer problem. If the reception problem is in phone lines, satellite antenna, or the wire service distribution box, receiving wire stories on two wireservers does not help.

Database Components for Backup Wire Distribution

Only one set of parallel and distribution servers are necessary, even if you are providing backup for up to ten wires. For backup wire distribution to work, distribution and parallel servers require certain database components, which are:

- A pair of scan queues to hold incoming wire stories for comparison by the parallel server—typically `SYSTEM.PARALLEL.A1` and `SYSTEM.PARALLEL.A2`
- A queue where the parallel server forwards unmatched stories for the distribution server to scan—typically `SYSTEM.PARALLEL.WIRES`
- A queue containing the wire distribution story for the distribution server, should it need to take over wire distribution—typically `SYSTEM.PARALLEL.DISTRIBUTION`
- Queues containing job lists used by the parallel and distribution servers located in the `SYSTEM.CONFIGURE` directory

Adding a Parallel Wire Server

A parallel wire server monitors two wire inputs of the same wire on different wireservers. If the primary source fails, the system will obtain stories from the secondary. For your parallel wire server to be truly redundant, each wire of the parallel server must be connected to an independent source of the incoming service.

Here, we assume that your site requires continuous reception of the AP wire, so we set up two wire inputs and a parallel wire server to receive stories from that wire service.

Additional details for the first two steps in the following procedure are provided in [“Adding a Server Program to the System”](#) on page 316.

To add a parallel wire server:

1. Choose a device and mailbox number for the server that will also be used for associated queues.
2. Add the server (utility program) to the configuration file on each iNEWS Server—such as servers A and B—in your system.

For instance: `server 258 parallel 258 - ;parallel server`

However, additional editing of the configuration file (`/site/config`) is required for parallel servers.

- a. You must create a line for the new wireserver and modify the existing wireserver. Change the device name AP to A1 to avoid name duplication:

```
wireserver 600 news A1 - ; primary wire from DR1
```

- b. Add the secondary wire:

```
wireserver 601 news A2 - ; backup wire from DR2
```

- c. When you finish making changes to the configuration file, save your changes and exit the line editor by typing what appears in bold:

```
w  
1795
```

```
q
```

Do not use an uppercase (W) in this step. When you press Enter, the editor responds by displaying a numerical value indicating the file size, which should be identical on all servers. See [“The Line Editor, ed”](#) on page 655 for more information.



Do not reconfigure or restart the wire programs yet. They do not match the codes in `SYSTEM.WIRES.DISTRIBUTION`, so all wires would be sent to `WIRES.UNKNOWN`.

3. Log in to an iNEWS Workstation. (You must create several queues and stories, as well as modify some that might already exist.)
4. Create a folder (directory) called Parallel in the System directory. The full pathname for this folder is `SYSTEM.PARALLEL`. See [“Creating a New Directory”](#) on page 102 for more information.
5. In `SYSTEM.PARALLEL`, create four new queues, giving them the following names: A1, A2, Wires, Distribution. The full pathnames for these queues will be:
 - `SYSTEM.PARALLEL.A1`
 - `SYSTEM.PARALLEL.A2`

- SYSTEM.PARALLEL.WIRES
- SYSTEM.PARALLEL.DISTRIBUTION

See “[Database Components for Backup Wire Distribution](#)” on page 354 for more information on the purpose for each of these queues. Also, see “[Creating a New Queue](#)” on page 104 for information on how to create a queue.



When wires are being distributed normally, the parallel server immediately removes matching stories from A1 and A2 queues. If the primary or secondary wire input stops reaching the A1 or A2 queue, automatic removal of stories from either queue stops. If the parallel server (or the NEWS it is on) dies, both queues continue to fill up until you run out of space. Finally, if the distribution server dies, SYSTEM.PARALLEL.WIRES fills up. To prevent these problems, set a short purge interval (less than six hours) for these three queues: A1, A2, and Wires. See “[Database Purge Intervals and Limits](#)” on page 139 for more information.

6. Assign mailbox database traits to three of the new queues, according to the following criteria:
 - SYSTEM.PARALLEL.A1 – shares same mailbox number as parallel server, such as 258
 - SYSTEM.PARALLEL.A2 – shares same mailbox number as parallel server, such as 258
 - SYSTEM.PARALLEL.WIRES – shares same mailbox number as distribution server, such as 257

Mailbox numbers must match the server’s mailbox number, as indicated in the configuration file. Otherwise, changes in the queue will not activate the correct server program.

- a. Open the Queue Properties dialog box for a queue, such as SYSTEM.PARALLEL.A1. See “[Changing Database Traits](#)” on page 136 for more information.
 - b. Click on the Maintain tab. See “[Maintain Tab](#)” on page 130 for more information.
 - c. Select the Standard radio button.
 - d. Type in the mailbox number, such as 258.
 - e. Click OK to save changes.
 - f. Repeat above substeps until queues have assigned mailboxes.
7. Create the parallel server’s job list queue in the SYSTEM.CONFIGURE directory in the iNEWS database. The queue’s name, such as 258, is the number selected earlier in this procedure for the parallel server. The queue’s full pathname is SYSTEM.CONFIGURE.258.



You can also end the name with a hyphen and descriptive word, such as “parallel,” in which case the queue’s full pathname would then be: `SYSTEM.CONFIGURE.258-PARALLEL`.

8. Create a job list story with tasks for the parallel server.
 - a. Open the job list queue, such as `SYSTEM.CONFIGURE.258`.
 - b. Create a new story—the first in the job list queue—to hold the parallel server’s job list. See “[Creating a New Story](#)” on page 111 for more information.
 - c. Add tasks that you want the parallel server to perform. Each task in the job list story consists of three lines, in the following format:

```
local <queue scanned by server>
remote <queue scanned by server>
distribution <destination queue>
```



The commands, `local` and `remote`, do not describe actual queue locations; both queues are on the same system.

| Command | Description |
|--------------|--|
| local | Identifies which queue contains wire stories forwarded to the parallel server through normal wire distribution. This job list command is similar to the scan command, used by action servers. The parallel server will scan the local queue and compare its contents to those in the remote queue. |
| remote | Identifies which queue contains the wire stories received from the secondary Data Receiver. This job list command is similar to the scan command, used by action servers. The parallel server will scan the remote queue and compare its contents to those in the local queue. |
| distribution | Specifies the destination queue where the parallel server forwards unmatched stories from the remote queue—the source queue scanned by the distribution server. |

For instance, the job list story may appear like this:

```
Parallel wire server 1/6/2010 13:19:35
local      system.parallel.a1
remote     system.parallel.a2
distribution system.parallel.wires
```



If the distribution server has not been set up, you must create it as well as its job list queue and story. See “Adding a Distribution Server” on page 350 for more information.

9. Modify the distribution server’s job list story.

For instance, go to SYSTEM.CONFIGURE.257-DISTRIBUTION and add the following lines to its job list story:

```
source system.parallel.wires
distribution system.parallel.distribution
```

10. Since you want the distribution server to distribute stories in the same way as standard wire distribution, copy the system’s wire distribution story (in SYSTEM.WIRES.DISTRIBUTION) to the SYSTEM.PARALLEL.DISTRIBUTION queue.



Your system’s wire distribution queue and story were included when your system was installed.

- a. Open the first story in SYSTEM.WIRES.DISTRIBUTION.
 - b. Select Edit > Copy To.
 - c. Enter SYSTEM.PARALLEL.DISTRIBUTION, the destination queue.
11. Back up the original wire distribution story in the SYSTEM.WIRES.DISTRIBUTION queue by making a copy of the story in that queue.
 12. Return to SYSTEM.WIRES.DISTRIBUTION to make the following modifications:

- a. Modify the bottom story (Leave the top copy unchanged, for now) so your system’s wire distribution sends a copy of every incoming wire story to the two queues scanned by the parallel server (SYSTEM.PARALLEL.A1 and SYSTEM.PARALLEL.A2) as specified in the parallel server’s job list story. To do this, add two lines (as shown) below the line that sends a copy of each wire story to WIRES.ALL. It should look like this:

```
AP##### wires.all ALWAYS URGENT
A1##### system.parallel.a1 ALWAYS
A2##### system.parallel.a2 TRANSMIT
```

- b. Modify every line in the wire distribution story that starts with AP, changing AP to A1. This makes it possible to tell whether a given wire story came from standard wire or parallel server distribution. When you are done, the three lines in the story should look like this:

```
A1##### wires.all ALWAYS URGENT
A1##### system.parallel.a1 ALWAYS
A2##### system.parallel.a2 TRANSMIT
```



Do not change the last line's A2 to A1. The A2 indicates the line that transmits stories to the SYSTEM.PARALLEL.A2 queue.

- c. Save the story.
 - d. Move the modified distribution story (at the bottom) to the top of the queue; this will make it the primary wire distribution story.
13. Connect the secondary wire to the secondary Data Receiver.
 14. Reconfigure the system.
 15. Start the original wireserver and the parallel and distribution servers.

For instance, when you see the `System being configured` message, start the servers with device numbers 257, 258 and stop wireserver 600 by typing what appears in bold:

```
NRCS-A$ restart 257 258 stop 600
```

```
NRCS-A$ stop 600
```

A message similar to the following will appear:

```
S258: Mon Jun 28 08:22:13 2010 matching system.parallel.a2
```

The message is from the parallel server and appears when it is started, indicating the server is alive and comparing the standard wire reception with wire stories routed to the A2 queue.

If primary reception fails, the parallel server gives the following message when it wakes up the distribution server:

```
S258: Mon Jun 28 08:22:13 2010 distributing system.parallel.a2
```

This message is the only way to tell when the parallel server is forwarding secondary wire stories for distribution. You can tell if these stories are being distributed by A2 instead of A1 by checking the `wiredistribution` codes in the stories.



There might not be any indication if secondary wire distribution fails. Check the local queue (SYSTEM.PARALLEL.A1) occasionally to see if it is accumulating stories. If so, this would indicate a problem with reception of the secondary wire.

If there is a problem with something other than the primary Data Receiver—for instance, if the wire fails in the phone or satellite connection—the redundancy provided by the parallel wire server cannot provide continuous wire service reception.

Keyword Servers

A keyword server is a utility program that searches stories it processes for specific keywords. It distributes copies of each story to different queues, based on which keywords the story contains. Unlike wire programs, which are limited to searches that contain no more than 250 unique keywords, a keyword server can perform searches that contain as many as 1000 unique keywords.

When you install a keyword server, it runs on one of your system's iNEWS Servers. While this lets the keyword server handle long lists of search rules, it also means the keyword server may affect your system's performance.

Suppose we have set up a wire program to perform keyword searching on stories it receives from the Associated Press (AP). Wire programs can only use 250 unique keywords in searches they perform.

To get around this limitation, a keyword server can be created to help the wire program, which still searches each story it receives using its own search rules. However, it also sends a copy of each story it receives to the keyword server, which in turn searches each copy using its own search rules. This effectively increases the number of unique keywords allowed to 1250: 1000 for the keyword server and 250 for the wire program.

An instruction is required in the wire distribution story that has the wire program duplicate a copy of each story it receives to a queue, which the keyword server watches. Some modification of the wire program's keyword job list queue (SYSTEM.WIRES.KEYWORDS) will also be required.

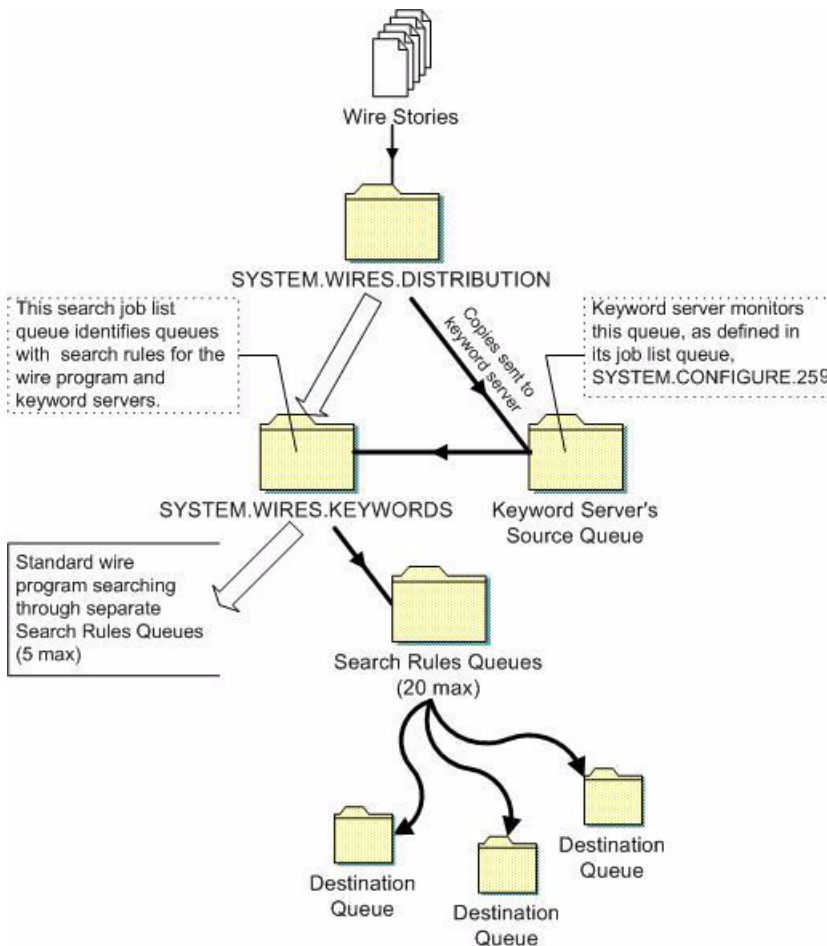
Database Components for Expanded Wire Keyword Searches

For expanded wire keyword searches to work, the keyword server requires certain database components. These components are:

- A queue containing job list used by the keyword server. It must be located in the SYSTEM.CONFIGURE directory.
- A source queue to hold incoming wire stories for scanning by the keyword server—such as WIRES.AP-SEARCH.
- A queue—typically named SYSTEM.WIRES.KEYWORDS—containing the search job list story. This queue is used by the system's standard wire searching feature; therefore, it should already exist in your database.
- Queues containing stories with search rule sets of keywords and destination queue names. The maximum number of search rule queues are 20 per keyword server.

- Destination queues where the keyword server forwards stories/search results.
- A queue containing the wire distribution story—typically named `SYSTEM.WIRES.DISTRIBUTION`. This queue is used by the system’s standard wire distributing feature; therefore, it should already exist in your database.

Here is a sample workflow diagram for a keyword server:



Adding a Keyword Server

Additional details for the first two steps and the last four steps in the following procedure are provided in [“Adding a Server Program to the System”](#) on page 316.

To add a keyword server:

1. Choose a device and mailbox number for the keyword server which will also be used for associated queues. A device name should also be chosen, such as key1; it will be needed for the next step in the procedure.
2. Add the server program (server line) to the configuration file on each iNEWS Server—such as servers A and B—in your system.



In addition to adding the keyword server to the standard host definition of the iNEWS Server on which you want it to run, ensure you also add it to each alternate host definition.

A keyword server's device name, such as key1, is used in the search job list; ensure that you always include a device name when adding a keyword server to the system. For instance, the configuration line might appear like this: server 211 keyword 211 key1 ;keyword svr

3. Create a job list queue in the Configure folder of the System directory in the iNEWS database. The queue's name, such as 211, is the same as the device and mailbox number selected earlier. For instance, the queue's full pathname is SYSTEM.CONFIGURE.211.



You should end the name with a hyphen and descriptive word, such as "key1." The queue's full pathname would then look like: SYSTEM.CONFIGURE.211-KEY1. See "Creating a New Queue" on page 103 for more information. In this example, the keyword server's device name is used as the descriptive word for the job list queue name. The device name is an important factor of later steps in this procedure.

4. Create the keyword server's job list story in the database, by doing the following:
 - a. Open the job list queue, such as SYSTEM.CONFIGURE.211-KEY1.
 - b. Create a new story, the first in the job list queue, to hold the keyword server's job list.
 - c. Add tasks that you want the keyword server to perform; include a separate task for each queue the server watches. Each task in the job list story consists of only one line, with the following format: source <queue scanned by server>

| Command | Description |
|---------|---|
| source | Identifies which queue the keyword server should scan. This queue is the same queue to which the wire program will be instructed to send wire story copies. |

For instance, here is an example of a job list story for a keyword server that scans WIRES.AP-SEARCH when it gets a wake-up notification:

| title | presenter |
|-----------------------|-----------|
| joblist for keyserver | |

```

;this server searches wires duped to wires.ap-search
source      wires.ap-search|

```

A single task is defined in the keyword server's job list, with a commented line—a line preceded with a semicolon (;)—offering further details about the keyword server.



When the keyword server processes a story, it deletes it from the source queue. You do not need to clear old stories out of the keyword server's source queue. It is recommended that you hide the source queue from general user accounts, by assigning it a read group, such as sysops.

5. Assign mailbox database traits, using the number matching the mailbox number chosen earlier in this procedure, to the appropriate source queue(s)—the queue(s) to be scanned by the server.

For instance, the job list queue is named 211, so that is the standard mailbox number assigned to queues identified in the job list associated with the distribution server.

- a. At an iNEWS Workstation, open the Queue Properties dialog box for a scan queue, such as WIRES.AP-SEARCH.
 - b. Click on the Maintain tab.
 - c. Select the Standard radio button.
 - d. Type in the mailbox number, such as 211.
 - e. Click OK to save changes.
6. Repeat the previous step as needed for each source queue in the job list.
 7. Create the queues and stories to hold search rule sets for the keyword server to use.



Each keyword server can have 1-20 search rules queues, each with its own keyword story, containing keywords. Each keyword story must be the first one in the queue, and contain a number of keywords equal to 1000 divided by the number of search rule queues. For instance, if four queues are set up for a keyword server, each queue's keyword story can hold

up to 250 keywords. This evenly distributes the 1000 keywords allowed for keyword servers across search rules queues and their stories. If 10 queues are set up, then each queue's keyword story can have up to 100 keywords.

- a. In this sample procedure, two search rules queues are created in the SYSTEM.WIRES directory. For instance, the queues' full pathnames could be SYSTEM.WIRES.KWDSERVER-1 and SYSTEM.WIRES.KWDSERVER-2.
- b. In each queue, the first story is considered the keyword story, and contains the rule set of keywords.

See [“Setting Up Wire Keyword Searches” on page 303](#), [“Keyword Limitations” on page 306](#), and [“Tips on Building Search Rules” on page 309](#) for more information.

The format each rule set must have is as follows:

```
destination <queue name>
<search rule>
[<search rule>]
[<search rule>]
. . .
```

| Command | Description |
|---------------|---|
| destination | Identifies the queue into which the keyword server will place the results of its keyword search. The name and location of this queue is customizable. |
| <search rule> | Each search rule consists of one or more keywords you want a keyword server (utility program) to use to locate stories on a particular topic. |



Never create a destination line that sends stories to a personal queue or any queue not purged. If a program sends stories to such queues, it can lead to an out-of-space condition.

Here is an example of what a keyword story and its rule set might look like:

```
destination      wires.keywords.drugs
panama AND drugs
drugs NOT pharmacy OR (drugs AND smuggling)
```

Each search rule occupies a separate line in the rule set. Also, the number of keywords allowed (1000 for keyword servers) is defined by the words, not the number of lines. For instance, in the previous example, four keywords are used—since you do not count each occurrence of a word (just the first), drugs is only counted once.



Each keyword in the story can have a maximum of 10 characters. To search on longer words, only use the first 10 characters. More examples and detailed information on rule sets and keyword searches are available in Chapter 12. See “Operating Wire Keyword Searches” on page 394 for more information.

8. Add an entry to the first story in the search job list queue, typically named SYSTEM.WIRES.KEYWORDS.
 - a. Go to that queue and open the first story for editing.
 - b. Insert an entry for the keyword server into the job list, which must begin with *20, followed by the keyword server’s device name. Use the same device name, which was defined in the configuration file during an earlier step, and the queues with the search rules, which were also created in a previous step of this procedure.

If you do not want the program to do any searching, that is all you need to put in the entry. If you want the program to search for keywords, follow the device name with one or more lines that list queues containing search rules you want the keyword server to use.

For instance, the keyword server has key1 as its device name and should use search rules in queues: SYSTEM.WIRES.KWDSEVER1 and SYSTEM.WIRES.KWDSEVER2.

Here is an example of how the story might appear:

| title | presenter |
|------------------|-----------|
| Keyword indirect | |

```

* A1 UPI
system.wires.keywords1
system.wires.keywords2
system.wires.keywords3
;
*20 key1
system.wires.kwdserver1
system.wires.kwdserver2
;
*
system.wires.default1
system.wires.default2
  
```



An optional destination prefix may also be included with the indirect list entries:

```
*20 <device name> [<destination prefix>]
```

If provided, the prefix is prepended to the destination queue of the destination line of the search rule.

9. Modify the wire distribution story, which is the first story in SYSTEM.WIRES.DISTRIBUTION. For the keyword server to search stories sent by the AP wire service, the wire program must duplicate a copy of each story it receives to the keyword server's source queue. For this to happen, the wire distribution story must be modified. So, open the story and add the new instruction on a single line, using the following format: `<dev name><wire code> <distribution queue> [<options>]`



There must be no spaces between the device name and the wire service code. The [<options>] should also be on the same line, separated from the distribution queue by a space.

| Code | Description |
|--|--|
| <code><dev name><wire code></code> | For instance, the wire program's device name is AP. If you want this instruction to match all wire stories regardless of their wire service code, use ##### as the code, because it matches any code. See "Distribution Name" on page 298 for more information. |
| <code><distribution queue></code> | The distribution queue is where you want stories with the specified code to be duplicated, such as WIRES.AP-SEARCH. |
| <code>[<options>]</code> | Finally, you can end the line with any of the following options: ALWAYS, SILENT, URGENT, FLASH, BULLETIN, or TRANSMIT. For instance, end the instruction with the ALWAYS option to ensure that every wire the AP wire program processes is duplicated to the keyword server's source queue. The finished instruction looks like: <pre>AP##### wires.ap-search ALWAYS</pre> |



See the table in "Notification Priority" on page 299 for more information on these options.

10. (Optional) Test your configuration changes.
11. Reconfigure the system.
12. Start the keyword server.

When you see the `System being configured` message, select the iNEWS Server where the keyword server is configured and start the keyword server. For instance, type:
`NRCS-A$ restart 211`
13. (Optional) Back up site files using `sitedump` command.

System Servers

Utility programs called system servers are usually included on your system at installation. You cannot customize them; therefore, they are described here in a limited way.

System servers include:

- Seek servers
- Fast Text Search (FTS) servers (Search and Index)
- Mail servers

The information in the following sections should be enough for you to maintain and use these servers, or add them if necessary.

Seek Servers

Seek servers allow users to search for text in the background, while users continue with other program activities, such as script writing. A seek server must be installed on your system.

The seek server program runs on the iNEWS Server, while you continue to work on your client computer (iNEWS Workstation). The seek server works by scanning the iNEWS database, story by story, going through the area you specified in the Find All dialog box. The seek server then returns stories resulting from its search as stories are encountered in the iNEWS database.

Seek search is used by anyone who uses iNEWS. The seek server can search any non-indexed queue or folder in your iNEWS database. If your site does not have a Fast Text Search (FTS) server, you will use the seek search exclusively.

Adding a Seek Server

This procedure involves editing the configuration file, which requires use of the line editor. See [“Adding a Server Program to the System” on page 316](#) and [“The Line Editor, ed” on page 655](#) for more information.

To set up the seek feature on your system:

1. Choose a device and mailbox number for the server program.
2. Add the server program's information to the configuration file.
 - a. The seek server's device number, such as 260, must appear in the servers line in the host definition for the iNEWS Server, such as server B, on which it runs. The line might appear as follows: `servers 257 258 259 260`



In addition to adding the seek server to the standard host definition of the iNEWS Server on which you want it to run, add it to each alternate host definition.

- b. Next, create a configuration line for the seek server at the bottom of the configuration file.
A seek server configuration line should look similar to this:
`server 260 seek 260 - ;seek server`
 - c. After adding information to the configuration file, save your changes.
3. Create the seek queue in the database.

The seek queue's name is defined in the Q_SEEK token in the /site/dict/queues file. Usually this name is SYSTEM.SEEK. Go to your System directory and see if this queue exists. If it does not, create it. See [“Creating a New Queue” on page 104](#) for more information.



This queue must exist for the SEEK command to work. Ensure the write group restriction is not applied so all users can use SEEK. See [“Write Group” on page 172](#) for more information.

4. Assign a form to the Seek queue, SYSTEM.SEEK.
A standard seek form was included with of your standard forms. The form is stored in SYSTEM.FORMS.S.SEEK. See [“Find All Form” on page 221](#) for more information.
For the SEEK command to use this form, assign it to the Seek queue using the Queue Properties dialog box and its Form tab. See [“Changing Database Traits” on page 136](#), [“Database Traits Summary” on page 122](#), and [“Forms Tab” on page 123](#) for more information.
5. Assign a mailbox to the Seek queue.

For the seek server to process a seek request placed in the Seek queue, your system must notify the seek server that a new seek request has arrived there. So, the Seek queue's mailbox number must match the seek server's mailbox number, as indicated in the

configuration file. Otherwise, changes in the queue will not activate the correct server program. Multiple seek servers can be configured. All seek servers should use the same mailbox, which is 260 in our example.

- a. At an iNEWS Workstation, open the Queue Properties dialog box for a queue, such as SYSTEM.SEEK.
 - b. Click on the Maintain tab.
 - c. Select the Standard radio button.
 - d. Type in the mailbox number, such as 260.
 - e. Click OK to save changes.
6. (Optional) Test your configuration changes.
 7. Reconfigure the system.
 8. Start the seek server.
For instance, type: `NRCS-A$ restart 260`
 9. (Optional) Back up site files using sitedump command.

Fast Text Search Servers

There are two server (utility) programs related to Fast Text Search (FTS): the one that performs fast text searches and the one that indexes stories in the FTS database. Both are explained in this section.

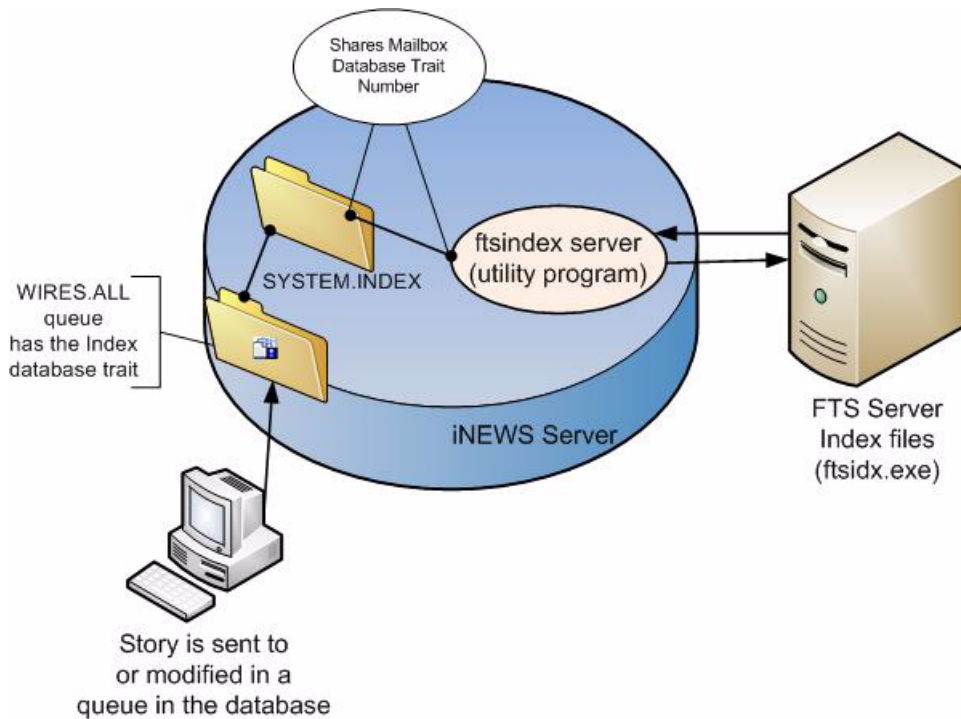
FTS is an alternative to the seek feature. FTS allows fast, efficient searching through large quantities of text material using indexes.



While both seek and FTS searches operate in the background—that is, the user can continue with other activities at the iNEWS Workstation while the system searches the iNEWS database—FTS searches will be faster than seek searches.

Fast text searching involves two separate activities: indexing data and searching the index for data. Different components are required for these separate activities to function.

The following figure shows the workflow iNEWS follows to create Index files of database stories, which can be queried later.



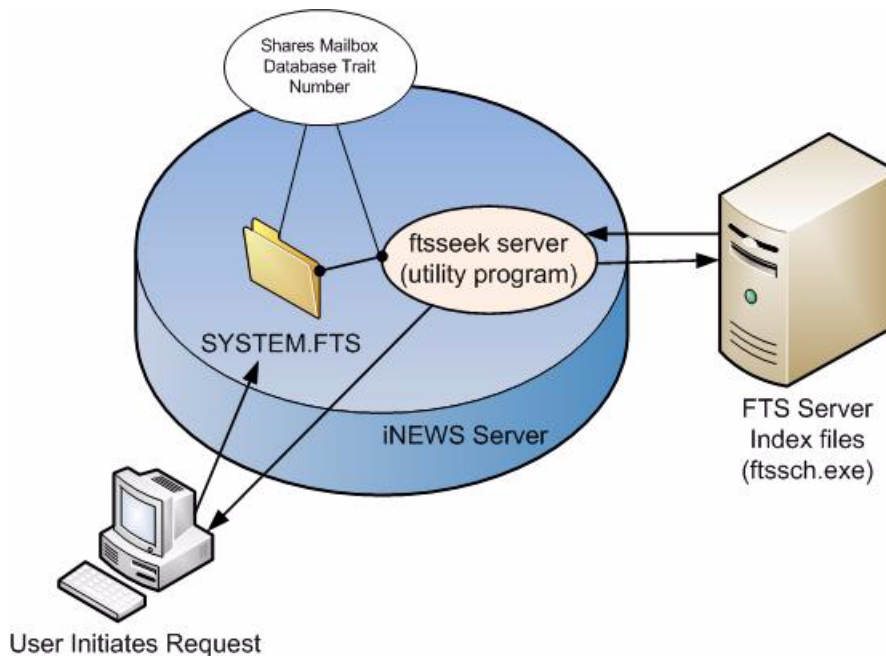
In a fast text search, the FTS server gets copies of material from those queues and folders in the iNEWS database marked with the Index icon (as shown in previous graphic). The system administrator identifies which queues and folders the system should index. See [“Batch Indexing” on page 377](#) for more information. The material in those queues and folders is fully indexed in the FTS server, so every word is added to an index file.

For each word, the index lists information about all stories where that word was encountered. The index is updated each time a story is added, changed, or deleted in an indexed queue. For instance, a story is added to WIRES.ALL. The system identifies that queue as an indexed queue and prompts the ftsindex server program to forward copies of the data to the FTS server for indexing.



The iNEWS system supports up to 50 separate index bases, 0-49. FTS handles the separation of the bases, but an administrator must assign index bases to directory records that should be placed in index bases other than 0 (zero), the default index base.

The following figure shows the workflow for a user's FTS request—the procedure iNEWS Workstation follows to search Index files and respond with results (pointers to database stories containing search criteria specified by the user).



In the above example, the user sends an FTS query to the iNEWS Server. The request is received in SYSTEM.FTS, which notifies the fsseek server program to conduct the search. (This is accomplished because both the SYSTEM.FTS queue and the server program share the same mailbox number.) The fsseek server program contacts the FTS server, which conducts a search of its Index files for data matching the user's search criteria, and then responds with pointers to those stories located in the iNEWS database. The fsseek server program forwards the FTS server's results to the user at the iNEWS Workstation.

Fast text search acts much like an index in a book. In other words, the fast text search stores references to where the information you search for is stored. For instance, if you were to search for the word Bush, the FTS server will note that the word appears in story 1, 8, 12, and 16. If you looked for the word Cheney, the FTS server will find the word in stories 2, 8, and 12. If you look for stories containing Bush & Cheney, you can view stories that appear in both lists (in this example, stories 8 and 12). This is faster than examining every story in the queue to see which contain both words. Instead, the system does that for you, and you benefit from the search.

If you invoke a background search (Find All or Find Global) for material in a folder or queue marked with the Indexed icon, iNEWS sends your search request to the FTS server. The FTS server will look up the words you specified in the index, and return a list of the resulting stories to the iNEWS Workstation.

Not all queues can be searched with the fast text search, simply because it takes space and effort to maintain an index. Therefore, system administrators will usually index only those queues searched often. You can determine which queues are available for fast text searches in three ways:

- In the Directory panel, the queue will display an Index icon as shown in the WIRES.ALL queue below:



- In the Find All dialog box, you will see an Advanced tab.
- In the Directory/Queue Properties dialog box, the directory/queue will have the index database trait selected.

Installing FTS Components on the Windows-based Server

FTS is an optional feature that requires certain configurations within the iNEWS system as well as additional hardware and setup. For instance, FTS requires the set up of two utility programs (ftsindex and ftsseek) on iNEWS Servers. Additionally, two executable files (ftsidx.exe and ftssch.exe) must be installed and running on a Windows-based server. The installation of these components are described here. For more information about the utility programs on the [iNEWS Servers](#), see [“Setting up FTS Components on the iNEWS Servers \(Linux\)” on page 374](#).

The Windows-based server running the two FTS executable programs (ftsidx.exe and ftssch.exe) is known as the FTS server. This computer is not to be confused with the two FTS utility programs (discussed in [“Setting up FTS Components on the iNEWS Servers \(Linux\)” on page 374](#)) known as ftsindex and ftsseek servers.



The FTS server should be running a Windows-based operating system. For information on the current operating system and service pack, contact Avid.

To install the executable files (FTS programs) required for FTS:

1. Insert the iNEWS DVD into the FTS server’s DVD ROM drive.
2. Navigate to the FTS folder on the DVD.

3. Double-click on the setup program (setup.exe) and follow instructions as they appear on screen.
4. The setup program will install several files, including ftsidx.exe and ftssch.exe. Make a note of the directory (folder) where they are installed.
5. A batch file is used to start the FTS processes (executable files installed in the previous step). The batch file is called go-fts.bat, and a shortcut is copied to the desktop.

The batch file contains two command lines in the code that you might need to modify based on the ports used by FTS. The batch file is in the installation directory. To allow editing of the file, right-click on the file and select Properties from the context menu. Then uncheck the Read-only attribute and click OK. To open the batch file for editing, right-click on the file and select Edit from the context menu. Locate the two lines in the code that look like this:

```
@cgi-fcgi -start -connect :6100 ftsidx.exe
@cgi-fcgi -start -connect :6101 ftssch.exe
```

The number 6100 corresponds to the Index *<port>* defined for W_BINDFTSI on the iNEWS Server, and 6101 corresponds to the Search *<port>* defined for W_BINDFTSS on the iNEWS Server. Ports 6100 and 6101 are defaults used by the FTS Index and Search programs. The W_BINDFTSI and W_BINDFTSS tokens are defined in the /site/dict/words file.

6. You must also create a directory (folder) on the FTS server that will be used by the Index program to hold index files it makes. You can create the directory using Windows Explorer.
 - a. Click Start.
 - b. Locate and select Windows Explorer in the Start menu.
 - c. Navigate to where you want the new directory to be and right-click.
 - d. Select New Folder from the context menu and name it, such as FTS.

For instance, the full pathname could be: C:\FTS



The token W_INDEXBASE in the /site/dict/words file must be defined to reference the directory created in this step. See “Setting up FTS Components on the iNEWS Servers (Linux)” on page 374 for more information.



The MSDOS command console window, which is opened by the batch file, must not be closed (although it may be minimized) after starting the ftsidx.exe and ftssch.exe programs. Closing the window will cause the two programs to exit.

7. Place a copy of go-fts.bat in the Startup folder for All Users.
 - a. Right-click on the Start icon on the Windows taskbar.
 - b. Select Explore All Users.
 - c. Open up Start Menu, Programs, Startup and place the copy there. This will ensure FTS starts whenever someone logs into the computer.

Setting up FTS Components on the iNEWS Servers (Linux)

The two utility programs used by FTS, known as `ftsindex` and `ftsseek` servers, must be set up on iNEWS Servers—that is, computers running the Linux operation system and iNEWS Server application, such as server A and server B.

To set up these utility programs, you may need to create some queues, such as `SYSTEM.INDEX` and `SYSTEM.FTS`, if they do not already exist in the iNEWS database. Also, some editing of files in the Site directory is required, particularly files in the `/site/dict` directory. To edit files in `/site/dict`, you will use `ed`, the line editor.

The following table shows tokens defined in the Queues and Words dictionary files in `/site/dict`. If the token name starts with a Q, the dictionary is in `/site/dict/queues`. If it begins with a W, it is defined in `/site/dict/words`.

| FTS Dictionaries | Explanation |
|------------------|---|
| Q_INDEX queue | Identifies the queue—typically <code>SYSTEM.INDEX</code> —that holds index requests from various server programs. The <code>ftsindex</code> program reads index requests from this queue, which is not visible on iNEWS Workstation. |
| Q_FTS queue | Identifies the queue—typically <code>SYSTEM.FTS</code> —that holds indexed search requests, much like <code>SYSTEM.SEEK</code> holds seek requests. |
| W_INDEXED | The word— typically set to <code>INDEXED</code> — used in indexed search requests to indicate that an indexed search is being requested. This type of search is similar to “fast” and “slow” seek searches. |
| W_INDEXBASE | Defines location of index files on the FTS server. Its format is <code><pathname></code> , which is a directory that exists on the FTS server. That directory contains a subdirectory for each index. The subdirectories have three-digit names with leading zeros. For instance, with <code>W_INDEXBASE</code> defined as <code>/C:/FTS</code> , the index base 0 would be in the directory <code>C:\FTS\000</code> and index base 32 would be <code>C:\FTS\032</code> . |

| FTS Dictionaries | Explanation |
|------------------|---|
| W_BINDFTSI | Identifies the binding address for the FTS server and its port, used for indexing. The format is <code><server>:<port></code> . The <code><server></code> is the hostname of the FTS server and <code><port></code> is the dedicated port number used for FTS search indexing. The default is <code>ftsserver:6100</code> where <code>ftsserver</code> should be replaced with the FTS server name and <code>6100</code> should be replaced with the number for the TCP port used by the <code>ftsidx.exe</code> program. If you alias the FTS server as <code>ftsserver</code> in <code>/etc/hosts</code> and used port <code>6100</code> , then no change is necessary. |
| W_BINDFTSS | Identifies the binding address for the FTS server and its port, used for searching. The format is <code><server>:<port></code> . The <code><server></code> is the hostname of the FTS server and <code><port></code> is the dedicated port number used for FTS text searching. The default is <code>ftsserver:6101</code> where <code>ftsserver</code> should be replaced with the FTS server name and <code>6100</code> should be replaced with the number for the TCP port used by the <code>ftsindex.exe</code> program. If you alias the FTS server as <code>ftsserver</code> in <code>/etc/hosts</code> and used port <code>6101</code> , then no change is necessary. |

To set up the `ftsseek` and `ftsindex` servers:

1. At an iNEWS Workstation, verify whether the Index and FTS queues exist in the System directory. If `SYSTEM.INDEX` and `SYSTEM.FTS` do not exist, you must create them.

2. At the console, add a definition for `Q_INDEX` to `/site/dict/queues`.

For example: `Q_INDEX /system.index`

3. Add a definition for `Q_FTS` to `/site/dict/queues`.

For example: `Q_FTS /system.fts`

4. Configure `ftsindex` and `ftsseek` server programs.

- a. Choose device and mailbox numbers for each server program.
- b. Edit `/site/config`, ensuring there is a server line for each server program.

For example:

```
server 261 ftsseek 261 - ;fts searches
server 262 ftsindex 262 - ;fts indexing
```

- c. Assign mailbox numbers to the appropriate queues in the System directory. The `ftsseek` server must be assigned the same mailbox that is assigned to `SYSTEM.FTS`. For instance, assign `262` to the queue defined for `Q_INDEX`, such as: `SYSTEM.INDEX`, and assign `261` to the queue defined for `Q_FTS`, such as: `SYSTEM.FTS`. Further details for assigning mailbox numbers (database traits) to queues are found in [“Assigning a Mailbox to a Queue” on page 328](#). Also, see [“Changing Database Traits” on page 136](#) for more information.

- d. Assign a queue form and story form to queues defined for Q_INDEX and Q_FTS (SYSTEM.INDEX and SYSTEM.FTS, respectively). The forms should be the same queue and story forms assigned to the queue defined for Q_SEEK, such as SYSTEM.SEEK. See “Assigning Forms to Queues” on page 390 for more information.
5. Add a definition for W_INDEXED to /site/dicts/words. It should be set to INDEXED.
For example: `W_INDEXED /INDEXED`
6. Add a definition for W_INDEXBASE to /site/dicts/words. Ensure the associated path exists on the FTS server and has appropriate access privileges. It should be an absolute path.
For example: `W_INDEXBASE /c:/FTS`
7. Set the binding address for indexing by adding a definition for W_BINDFTSI to /site/dict/words. Use the format: `<server>:<port>`
For example: `W_BINDFTSI /dexy:6100`
In the example, the FTS server has a hostname of “dexy.”
8. Set the binding address for searching by adding a definition for W_BINDFTSS to /site/dict/words. Use the format: `<server>:<port>`
For example: `W_BINDFTSS /dexy:6101`



You must define separate port numbers for W_BINDFTSI (indexing) and W_BINDFTSS (searching) on the FTS server.

9. (Optional) Assign index bases, using the dbtraits command in the following format:
`dbtraits <dir> ftsindex <base>`
Values for <base> can be any numerical value from 0-49. All directories default to index base 0 (zero).
For example:
`dbtraits wires ftsindex 1`
`dbtraits archive.2008 ftsindex 2`
`dbtraits archive.2009 ftsindex 3`



The list d-i command displays at the console 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.

To start the ftseek and ftsindex servers on the iNEWS Server:

- ▶ Use the restart console command in the following format:
`restart <device number> <device number>`

For example: select the appropriate server at the console, and type: `restart 261 262`

To stop the `ftsseek` and `ftsindex` servers on the `iNEWS` Server:

- ▶ Use the stop console command in the following format:

```
stop <device number> <device number>
```

For example, select the appropriate server at the console, and type: `stop 261 262`

To stop the FTS programs on the FTS server from the console:

- ▶ Type:


```
ftsdebug index shutdown
ftsdebug search shutdown
```

Batch Indexing

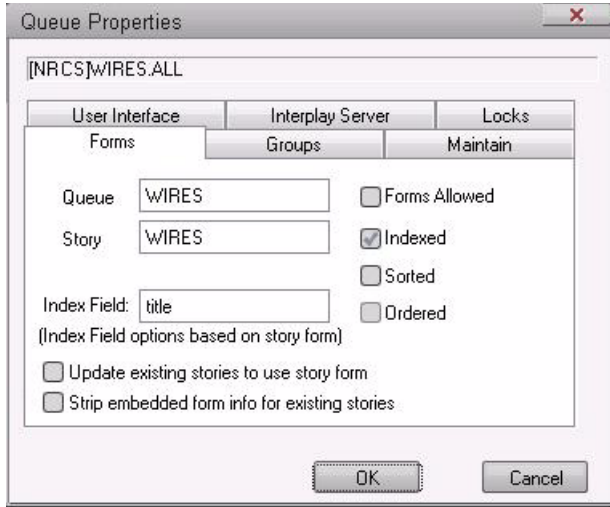
Batch indexing is a way for a system administrator to define queues that should be indexed or de-indexed, and cause an update of the index base to incorporate the current state of the queues' stories in the index base.




An index base is the set of four index files on the FTS server. The iNEWS system can maintain up to 50 index bases in the folder defined by the `W_INDEXBASE` token in `/site/dict/words`.

Batch indexing can be performed at either the queue or directory level by applying the index database trait. See [“Changing Database Traits” on page 136](#) and [“Database Traits Summary” on page 122](#) for more information.

For instance, if you want to mark a queue as indexed and schedule an index base update, you can apply the Indexed database trait to the queue using the Queue Properties dialog box.



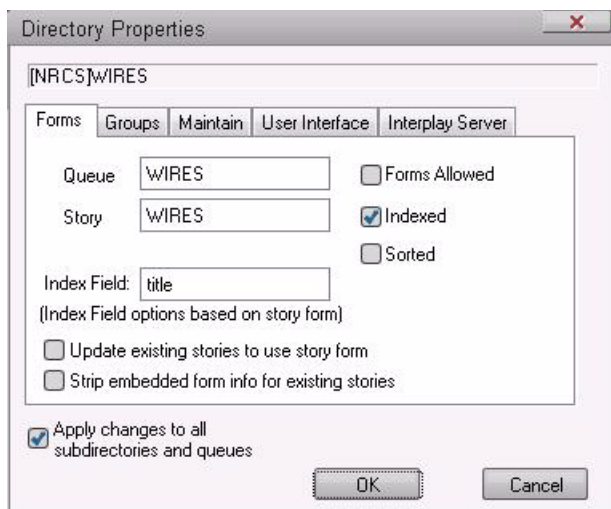
This command marks the chosen queue as indexed and submits a request to update entries in the index base for all stories in the queue.

 When indexed, the queue will appear with the Indexed icon (shown at left) in the Directory panel of the iNEWS Workspace.


Similarly, you can uncheck the Indexed check box in the Queue Properties dialog box if you want to mark a queue as deindexed and schedule an index base update. This command marks the chosen queue as not indexed and submits requests to remove all story entries for the queue from the index base.


Directories (Folders)

Batch indexing can be applied to multiple queues in a directory. For instance, you can mark all queues at every level under a directory (folder) as indexed and schedule an index base update by applying the index database trait to the directory, using the Directory Properties dialog box. Similarly, to mark all queues at all levels under a directory as deindexed and schedule a de-index, uncheck the Indexed check box in the Directory Properties dialog box.



The ‘Apply changes to all subdirectories and queues’ check box must be checked to have the change propagated to all subdirectories and queues of the directory.

 When indexed, the directory will appear with the Indexed icon (shown at left) in the Directory panel of the iNEWS Workspace.

 *The database traits program will ignore requests to index or deindex the queues defined for the dictionary entries `Q_SEEK`, `Q_FTS`, `Q_INDEX`, and `Q_DEAD` in `/site/dict/queues`.*

Dynamic Indexing

Dynamic indexing is a function of the iNEWS Server that automatically triggers an index change when a story is added, changed, moved, or deleted. If you set index database traits for queues to be indexed, the iNEWS Server will automatically schedule updates to the index base whenever a story in an indexed queue changes.

Archival and Backup

If you want to save the current state of an index base, you must copy the Index files to another location and then archive/backup the files as you would normally do for a Windows-based server system. (See Microsoft’s server documentation for details on backing up and archiving files.)

 *Archived index bases might be out of sync with the active iNEWS database.*

Reindexing (Optional)

If an index base becomes unusable, you should reindex.

To reindex:

- ▶ At the console, issue a `dbtraits` command with the `reindex` option for one or more folders that might contain indexed queues. This will traverse the folder's hierarchy and schedule index requests for each indexed queue found.

For instance, type: `dbtraits wires reindex`

This will cause the system to revisit all queues anywhere under the Wires directory and schedule indexed queues for indexing.



The `ftsdebug` command may be used to troubleshoot communication between the iNEWS Servers and FTS programs, compare contents of indexes and indexed queues, etc. For more information on this command, see the iNEWS Newsroom Computer System Operations and Troubleshooting Manual.

Mail Server

Your system provides a mail feature that allows users to send mail messages to other users. To enable the mail feature, a utility program known as the mail server must be installed on your system. Generally, this is done when your system is installed.

Disabling Mail to All Users

Ordinarily, a user can send mail to all, which sends a copy of the mail story to every user on the system. If you want to disable this feature, include an alias for all in a story in `SYSTEM.GROUPS` that maps the word all onto a nonexistent user name. The new alias may look like this:

```
alias all
not-allowed
```

Because not-allowed is not a valid user on your system, anyone who tries to send mail to all will get the error message: `Returned mail: User unknown.`

Network Mail

Your system has a hosts file that it uses to communicate with other devices on the network. This file is found in `/etc/hosts`. Each iNEWS Server's hosts file is like a directory that includes the network addresses of all computers on the network. The iNEWS Servers use that directory to find the address of other computers. See [“Hosts File” on page 254](#) for more information.

The typical format of a line in the hosts file lists the Internet address of a computer or other network device, followed by its name.

For instance, here is an `/etc/hosts` file for a three-server system:

```
10.0.0.1 nrcs-a a
10.0.0.2 nrcs-b b
10.0.0.3 nrcs-c c
```



Some sites may have DNS, not just a hosts file. E-mail servers will refuse e-mail from iNEWS if it comes from a non-verifiable domain, such as nrcs-a, so to ensure everything works correctly, check network settings at the site and on e-mail servers and put the iNEWS Server's FQDN as the first entry against its IP address in `/etc/hosts`.

Monitor Servers

Monitor servers are utility programs—running on the iNEWS newsroom computer system—that check shows' event requests for errors, create composite and event lists, and send playlists to Command, ControlAir, or MOS Gateway.

A monitor server must be assigned device and mailbox numbers—typically, these numbers are the same, which makes the configuration easier to remember. A device number is chosen for a monitor server so iNEWS recognizes the server (utility) program as a valid device. The device number must be entered in the system's configuration file. The mailbox number must be assigned to both the show's monitor server and its rundown queue.

A mailbox is an activation mechanism for a server (utility) program, so if a queue has a mailbox number matching a server program, then that server is the one activated or “awakened” whenever something happens to the queue.

The mailbox enables iNEWS to notify the monitor server of changes made to a rundown queue—the one sharing the same mailbox number—while that show is monitored. The monitor server then updates composite and event lists, if necessary.

The monitor server is turned on when a user at an iNEWS Workstation turns it on. When on, monitor server checks its assigned rundown for machine control events, and builds playlists. It continues to monitor the rundown for changes and performs as much error checking as it can—by comparing machine control events with styles located in the `SYSTEM.RESOURCE` queue—without communicating with any broadcast equipment, including Command. To learn more about styles and how monitor servers check for errors, see [“Creating Styles” on page 400](#).

The monitor server loads data to the Avid iNEWS Command system or the MOS Gateway system when a user instructs it to do so from an iNEWS Workstation.

13 Servers



It is recommended that producers turn the monitor server on early so it can check the rundown queue throughout much of the show's development.

Checklist: Monitor Server Configuration

When configuring monitor servers, there are certain tasks that must be performed on the iNEWS Servers at the console, and other tasks at an iNEWS Workstation. These tasks must be completed to download playlists and updates to the playout system.


This section assumes:

- Your site has a fully functioning network.
- Your newsroom computer system servers are operational and running the iNEWS Server software.
- All iNEWS Servers are connected to a console multiplexor, to which Avid technicians have dial-in access.
- The person performing the installation has attended an iNEWS system administration course or has equivalent experience. This includes having a working knowledge of ed, the line editor and selecting servers at the console.

The following steps do not include setting up external devices, such as character generators, which play broadcast events. Refer to the manufacturer's documentation provided with the device for that information. For more configuration details, other resources include: the *Avid iNEWS Command Installation and Configuration Guide* as well as *Device Manager Guides*, provided on the ControlAir installation CD.

Complete the following tasks in the order that they are listed. If you are viewing this checklist online, you can easily link to a topic that provides procedural details for the task. Return to this main checklist after you have completed each task.

| ☒ | Tasks | Refer to... |
|---|--|---|
| ☐ | Create a monitor server for each show. | “Creating a Monitor Server for Each Show” on page 384 |
| ☐ | Create composite and event list queues. | “Creating Composite and Event List Queues” on page 387 |
| ☐ | Set up queue and story forms. | “Set up Queue and Story Forms” on page 390 |
| ☐ | Assign forms to queues. | “Assigning Forms to Queues” on page 390 |
| ☐ | Create an entry in the SYSTEM.MAP story. | “Creating an Entry in the SYSTEM.MAP Story” on page 391 |

| ☒ | Tasks (Continued) | Refer to... |
|---|---|--|
| ☐ | (Optional) Update the iNEWS system dictionaries. | “Updating the iNEWS System Dictionaries” on page 397 |
|  | <i>Some configurations require updating styles in the SYSTEM.RESOURCE story. For more information on how to do that, see “Creating Styles” on page 400.</i> | |

Creating a Monitor Server for Each Show

This section assumes rundown queues for the shows being monitored already exist. If not, create them before continuing.

To create and configure a monitor server:

1. Check the configuration file (/site/config) and choose the next available device number for the monitor server, from the range of numbers reserved for use by your system’s server programs, such as 201 to 300.
2. Choose a mailbox number. Although valid standard mailbox numbers are 1 through 5000, for simplicity, this number is usually set to the same device number.

Remember, you must ensure no other device is using the chosen mailbox number by using a variation of the list c command at the console.

For instance, to check whether a specific mailbox number, such as 267, is assigned to a server program, type: `list mailbox=267 c`

Information similar to the following appears:

```
DEV DEVICE_TYPE COMPUTER NOTIFY OPTIONS DEVNAME
```

If you see the device configuration header with no information below it (as shown), then no device has that mailbox, and you can use that number. However, if configuration information for a device appears below the header, that device has the same mailbox as the one you chose. Therefore, choose another mailbox number.



If the mailbox number is already being used, and you must choose another, you might want to try another device number for the monitor server as well. Typically, mailbox numbers match utility program device numbers, although this is not required. However, the mailbox number you select must also be assigned to the queue monitored by the monitor server (utility program), a step which is covered later in this procedure.

You can also use a variation of the list command to view all mailbox numbers assigned to monitor servers or other utility programs and devices. See [“Viewing Information about Devices” on page 263](#) for more information.

3. Add the monitor server to the /site/config file on each iNEWS Server—such as server A and server B in a dual server system—by doing the following:



Always back up the /site/config file before making any changes. See “Editing the Configuration File” on page 252 for more information.

- a. Select all servers.
- b. Open and edit the configuration file, by typing (what appears in bold):

```
NRCS-A$ ed /site/config
1259
```

After you press Enter, the editor responds by displaying a numerical value indicating the file size expressed as the number of characters, including spaces and returns.



This procedure, which modifies the /site/config file uses ed, the line editor. If you do not know how to use ed to modify lines in the file, please see “The Line Editor, ed” on page 655.

- c. Add the monitor server’s device number to the servers line in the host definition for the iNEWS Server that will run the monitor server program.

For instance: `servers 261 263 265 267`

The device number 267 is added to the servers line in this example.



Divide your server programs evenly among your iNEWS Servers to distribute the load they put on your system. For instance, put odd numbered programs on server A and even numbered ones on server B. Additionally, ensure that you also add the configuration line for the monitor server to alternate host definitions for your iNEWS Servers. This ensures it can run on the surviving computer should one of your iNEWS Servers stop functioning.

- d. Add a configuration line for the monitor server in the host definition belonging to the iNEWS Server that will run the server program. This line begins with the word server and contains the mailbox number assigned to the monitor server.

The format for server programs’ configuration lines are:

```
server <device #> <type> <mailbox #> <device name>
```



Do not confuse the configuration line in this step, which starts with server, and the servers line mentioned in the previous step, which lists device numbers. They are not the same.

| Parameter | Description |
|-----------|---|
| device # | The device number assigned to the server program. This number must also be listed in the servers line in a host definition. |
| type | The type of server program is monitor. |

| Parameter | Description (Continued) |
|-------------|--|
| mailbox # | The mailbox the server program uses. This number typically matches the monitor server's device number. |
| device name | Monitor servers do not use a device name; enter a hyphen (-). |

The following are sample configuration lines for monitor server programs:

```
server 264 monitor 264 - ;monitor server
server 265 monitor 265 - ;monitor server
server 266 monitor 266 - ;monitor server
server 267 monitor 267 - ;monitor server
```

Comments appearing after the semicolons (;) are optional.

- e. When you finish making changes to the configuration file, save your changes and exit the line editor by typing (what appears in bold):

w

1279

q

Do not use an uppercase (W) in this step. When you press Enter, the editor responds by displaying a numerical value indicating the file size. See [“The Line Editor, ed” on page 655](#) for more information.

4. (Optional) Test your configuration changes.
5. Reconfigure the system. When adding server programs, you do not need to stop anything to reconfigure the system.



Unlike other utility programs, it is not necessary to start a monitor server when you add it to the configuration file. It starts when someone uses the monitor on or monitor load command for the rundown queue to which you have assigned the monitor server. See [“Using the Monitor Server” on page 405](#) for more information.

6. (Optional) Back up site files using sitedump command.
7. After you've chosen the monitor server's mailbox number and verified that it is not used by another device, you must also assign it to the show's rundown queue, being monitored by the monitor server.

For instance, if the monitor server's mailbox number is 267 and it will monitor the SHOWS.6PM.RUNDOWN queue, then that number must be assigned to that queue. This step is done at an iNEWS Workstation, using the Queue Properties dialog box. See [“Assigning a Mailbox to a Queue” on page 328](#) for more information.



Because multiple queues can share the same mailbox number, care must be taken to avoid conflicts. Rundown queues should never share the same mailbox number.

Since multiple queues can share the same mailbox, you can use a variation of the `list d` command at the console to list all queues and directories in the database that are using a certain mailbox, such as 267. To do this, type: `list mailbox=267 d`

Creating Composite and Event List Queues

The monitor server scans stories for machine control events and builds lists of these events. These lists, known as composite and event lists, are optional, yet valuable, resources for a show's director and production device operators. An event list contains details for a specific device, while a composite list contains status information for all devices connected.

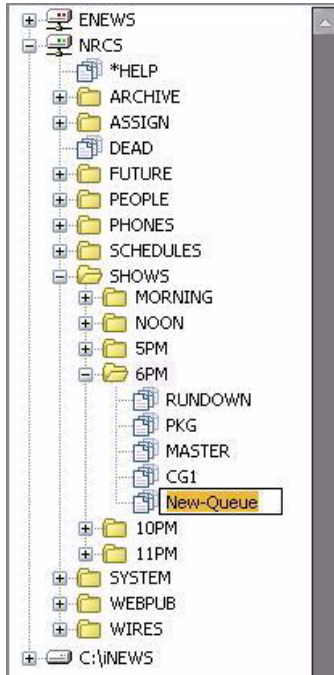
You might want to create only composite and/or event lists for certain production devices. If you do not create list queues for some devices, the monitor server will still function normally.

For these lists to exist, queues for them must be created—if they do not already exist—and they must be “mapped” to the monitor server so it knows where to put composite information and event lists.

The following procedure explains how to create queues. For more on mapping them, see [“Creating an Entry in the SYSTEM.MAP Story” on page 391](#).

To create and configure the composite and event list queues:

1. At an iNEWS Workstation, create the queues in the iNEWS database by following the procedure given in [“Creating a New Queue” on page 104](#)
2. Name the new queues, such as COMPOSITE and CG1.



The pathname of each device's event list queue is a combination of the event list directory and the device manager's name. For instance, if you want an event list for the 6PM show that has a character generator called CG1, you may select to place the event list queue in the SHOWS.6P directory. The pathname would be SHOWS.6P.CG1. Using the same example, the pathname for the composite list queue would be SHOWS.6P.COMPOSITE.

The newly created queue will inherit database traits of its parent directory initially. You can open the new queue by double-clicking on it. For more information about queues and their database traits, see [“The Database: Directories, Queues, and Stories”](#) on page 99.

3. Set the display of the composite and event list queues.
 - a. Log in as a system administrator at an iNEWS Workstation.
 - b. Navigate to and right-click on the composite or event list queue in the Directory panel.
 - c. Select Properties from the context menu.

The Queue Properties dialog box appears.



Access to the Directory/Queue Properties dialog box and its appearance vary, depending on certain circumstances. See “Database Traits Summary” on page 122 for more information.

- d. Select the User Interface tab.
 - e. Set Preview Lines to the number of lines you want displayed in the Queue panel.
To have monitor server display information in these lists properly, each queue must have a 1-line display and be assigned the proper queue form. Assigning these forms is covered in “Assigning a Form as a Queue or Story Form” on page 208.
 - f. Click OK to save changes.
4. Ensure that you can use the queues effectively by removing the inverted database trait and (optionally) applying the refresh database trait to them. This can be done by removing the check mark from the Inverted checkbox and selecting the Refresh checkbox on the User Interface tab of the Queue Properties dialog box at any iNEWS Workstation.

For more information about assigning database traits to queues, see “Changing Database Traits” on page 136.

5. Assign a write security group to your event and composite list queues to ensure that only the monitor server makes changes to the composite and event lists. It is recommended that you restrict writing access of these queues to superusers.

For more information on how to assign write groups to queues, see “Group Traits for the Database” on page 170.

6. On the Maintain tab, set the queue to purge in 18 hours, select the Skip Backup checkbox, and set Save Old Versions to Save None.

After the composite and event list queues are created, the monitor server must be created—as described in [“Creating a Monitor Server for Each Show” on page 384](#)—and mapped to them. This will enable the monitor server to determine the queue where the composite list should be placed and the directory where the event list queues are located. The procedures for mapping the monitor server to composite and event list queues are explained in [“Creating an Entry in the SYSTEM.MAP Story” on page 391](#).

Set up Queue and Story Forms

All three queues—rundown, composite, and event list—use forms with fields that may be filled in by monitor server programs.

Add fields to existing rundown queue and story forms in iNEWS, as well as forms used by the composite and event list queues, using the following guidelines:

- Two fields are used in forms for all three queues: VIDEO-ID and EVENT-STATUS.
- Two fields are used in forms for rundown queues: STATUS and TAPE-TIME.
- The following fields are used in forms for composite and event list queues: MOS-TITLE, CG-ADDR, ITEM-CHANNEL, STILL-ID, CG-TEXT, CG-TEMPLATE, STILL-PRESET, STYLE, and EFFECT.



For a field to exist in a queue form, it must also exist in a story form. See [“Forms” on page 197](#) for more information about forms and field types.

Assigning Forms to Queues

After you create queues to hold the composite and event lists, you must assign forms containing iNEWS fields to each queue. For instance:

- Assign a form designed to display composite list information to SHOWS.6PM.COMPOSITE.
- Assign a form designed to display the information in a character generator’s event list to SHOWS.6PM.CG1.



Your system includes default forms—located in the SYSTEM.FORMS directory—for composite, character generator, and video event list queues. If upgrading to iNEWS from a previous product version, you must update forms for composite and event list queues, including rundown forms.

To assign a form to a queue:

1. At an iNEWS Workstation, navigate to the queue you want in the Directory panel.
2. Right-click on the queue.
3. Select Properties from the context menu.

The Queue Properties dialog box appears.



Access to the Directory/Queue Properties dialog box and its appearance vary, depending on certain circumstances. See “Database Traits Summary” on page 122 for more information.

4. Do either or both of the following:

- ▶ Use the Queue drop-down list on the Forms tab to select the form you want to apply to the queue as the queue form database trait.
- ▶ Use the Story drop-down list on the Forms tab to select the form you want to apply to the queue as the story form database trait.

You must assign a queue form and a story form. The queue form determines the look of the queue. The story form determines how an event request is displayed when you double-click it.

For instance, to assign the standard composite list queue form to the SHOWS.6PM.COMPOSITE queue, select the appropriate form from the Queue drop-down list in the Queue Properties dialog box. To assign a story form to this composite list, select the appropriate form from the Story drop-down list in the Queue Properties dialog box.



The forms will only appear in the drop-down lists if they exist in the database. For more information about the Forms tab and how to create your own forms or modify existing forms, and about database traits, see “The Database: Directories, Queues, and Stories” on page 99 and “Forms” on page 197.

5. If you made changes to an existing form, you must select the Update existing stories to use story form. When this check box is selected, iNEWS changes the story form assignment for existing stories with the queue.
6. Click OK to save changes and apply the new queue/story form settings.



Users should log off and sign back on to view the new queue/story form settings.

Creating an Entry in the SYSTEM.MAP Story

The map story is a standard iNEWS database story and is always the first story in the SYSTEM.MAP queue (as defined by Q_MMAP). It can be opened and edited like any other iNEWS database story; however, access to it is typically limited to system administrators who already have access to the System directory.

When you create a show’s map story entry, you might want to specify that monitor server create and maintain event and composite lists when someone monitors the show.

After creating the show’s monitor server, add an entry for the show to your system’s map story. This entry specifies to the show’s monitor server the location of the show’s rundown queue, and composite and event lists. Without this information, the monitor server will not create lists. The map story entry also specifies the list of groups that can monitor a queue, and when the monitor server turns itself off.

To add an entry for the show in the map story:

1. Open the SYSTEM.MAP story, which contains a separate entry for each show that will be produced using Command, ControlAir, or MOS Gateway.
2. Create an entry header for the show.

Each show’s SYSTEM.MAP story must begin with a line called the entry header. The format for the entry header is:

```
<RD queue> <EL directory> <CL queue> <monitor grp> <mon off time or duration>
```

| Parameter | Definition |
|-----------------|---|
| <RD queue> | This is the show’s rundown queue. Enter the full pathname. |
| <EL directory> | (Optional) This is the event list directory. Enter the directory in which event lists are stored. The monitor server combines information you put here with the device name. For instance, if you specify SHOWS.6PM here and you have a CG machine called “cg1,” the monitor server puts the character generator’s event list in shows.6pm.cg1. It is a good idea to use the same directory that holds the show’s rundown. If you do not use this field, put a dash here. |
| <CL queue> | (Optional) This is the composite list queue. Enter the full pathname of the queue in which the show’s composite list is stored. It is a good idea to put the composite list in the same directory as the show’s rundown queue. If you do not use this feature, put a dash here. |
| <monitor group> | (Optional) Put a security group in this field so that only superusers and people assigned to that group can monitor the show. Put a dash here if you do not want to restrict who can monitor the show. Consider using different security groups for each show in a series of back-to-back shows to prevent one show’s producer from starting or stopping another show’s monitor server. |

| Parameter | Definition (Continued) |
|----------------------------|--|
| <mon off time or duration> | <p>The time you want the show's monitor server to turn itself off—when you want the system to stop monitoring the show.</p> <p>You can enter this time as either a specific time of day or a duration.</p> <ul style="list-style-type: none"> • Enter the time of day in 24-hour format. For instance, type 1915 to have the monitor server turn itself off at 7:15 PM. • Enter a duration by typing D or d before a numeric value. An uppercase D denotes a duration in hours, and a lowercase d denotes a duration in minutes. |

The following lines are a couple of examples of entry headers, one with and one without a monitoring security group defined:

```
show.noon.rundown  show.noon  show.noon.composite  -          1245
show.6pm.rundown  show.6pm  show.6pm.composite  monitor  D2
```

3. Enter the information line for a server.

The entry header is followed by an information line in the following format:

```
<Server Type> <Server Name> <Backup Server> <CAWS form>
```

| Parameter | Definition |
|-----------------|---|
| <Server Type> | <p>The options for server type are:</p> <ul style="list-style-type: none"> • WNASVR - used for a Command Server • CASVR - used for a ControlAir Server • MOSSVR - used for a MOS-based server, such as one running the MOS Gateway software. |
| <Server Name> | The server name is the network resolvable address of the server machine. |
| <Backup Server> | The backup server field may contain a placeholder if no backup server is configured. The backup server can be loaded in place of the primary server; however, both may not be loaded at the same time. |
| <CAWS form> | CAWS stands for ControlAir Workstation. If the CAWS form field contains a placeholder, the story form assigned to the composite list queue will be used. If no composite list queue is configured, the story form assigned to the rundown queue will be used. For best performance, a CAWS form should be created and assigned here. That form should contain only the title and page-number fields, and any event-related fields such as VIDEO-ID, MOS-TITLE, etc. |

- List the devices—production devices for which you want the show’s monitor server to process event requests.

The monitor server can only process event requests for devices specified in the device list; event requests for other devices generate errors.

Each line in the device list contains information for each device manager configured on that server. All device information lines share three required fields: device type, device name, and alias name, followed by device-dependent information.

For Command Servers (wnasvr), the format for video, CG devices are:

```
video <device name> <alias name> <default channel>
cg    <device name> <alias name> <default channel> <style>
```

For ControlAir Servers (casvr), the format for video, CG, and MOS devices are:


```
video <device name> <alias name> <channel assignment agent>
cg    <device name> <alias name> <msg drive>:<msg dir> <tmpl
drive>:<tmpl dir> <start address> <end address> <address offset>
mos   <device name> -
```

| Parameter | Definition |
|---------------|---|
| <device type> | <p>The options for device type are: video, cg, or mos.</p> <p>In some cases, iNEWS users may create machine control events that specify devices by type rather than name. When the monitor server encounters such an event, it sends it to the first device of that type defined in the show’s SYSTEM.MAP story entry. If you have a show that uses two devices of the same type, decide which device you want as the monitor server’s default, and define it first in the show’s SYSTEM.MAP story entry.</p> |



The iNEWS newsroom computer system stores a Machine Control System (MCS) dictionary file—named mcs—in the /site/dict directory. The /site/dict/mcs file usually defines production device types as CG for character generator and VIDEO for video servers. You must specify in the device list a valid production device as defined in the dictionary file; otherwise, the monitor server ignores that line and any that follow it, so it cannot properly process event requests.

| Parameter | Definition (Continued) |
|----------------------------|---|
| <device name> | <p>Specify the name of the production device in this parameter.</p> <p>Device names for Command Server devices must match the channel group name in Command. See the <i>Avid iNEWS Command Administration Guide</i> for more information.</p> <p>Device names for ControlAir Server devices must match the device name specified in the DMP file for one of the device managers configured on the ControlAir Server. See the <i>Avid iNEWS ControlAir Operations Manual</i> for more information.</p> |
| <alias name> | <p>The alias name field may contain a placeholder if an alias is not used. A device alias name provides a means to enter generic production cues into stories which may be resolved to different device names when those stories are copied to different rundown queues. The default alias name for any device manager is the device type. When a production cue is processed, the device name in the cue is compared to the device list for that rundown. If no match is found, the alias list is searched. If a match is found in that list, the production cue is assigned to the corresponding device manager. Duplicate device names are not allowed. However, alias names are not checked for uniqueness. If an alias applies to two or more devices, the first in the list will be selected.</p> |
| <channel assignment agent> | <p>This parameter specifies whether the iNEWS system or the ControlAir Workstation assigns channels. Use these codes:</p> <p>1 to have channel assigned by iNEWS</p> <p>3 to have channels assigned by ControlAir Workstation</p> |
| <msg drive>:<msg dir> | <p>Enter the disk drive and directory you want the machine to use.</p> <p>CG drive and directory fields still use a colon (:) to separate two components. A placeholder (-) or single colon will cause both to be empty. If you leave this field empty, the machine uses its default drive.</p> |
| <tmpl drive>:<tmpl dir> | <p>CG devices can specify a separate drive and directory where template messages are stored on the CG. A placeholder (-) or single colon will cause both to be empty. If you leave this field empty, the machine uses its default drive.</p> |

| Parameter | Definition (Continued) |
|-------------------------------|--|
| <start address> <end address> | <p>Specify a range of addresses in these fields. The monitor server uses this range to store the forms it builds.</p> <p>To specify a range of addresses, enter the starting address, followed by a space and the ending address.</p> <p>For instance, to reserve addresses 1 through 199, type: 1 199.</p> <p>The range of numbers must be large enough to hold all character-generated graphics—also known as supers—that monitor server is likely to build for the show. Also, ensure you do not select a range that conflicts with the addresses the character generator uses to store its permanent supers or CG forms.</p> <p> <i>Because the monitor server may skip an address to keep them contiguous within a story, this range should be larger than the largest number of possible addresses used.</i></p> |
| <address offset> | <p>The CG address offset field is optional. If entered, it is the number of CG addresses to skip when a new story is parsed before assigning new addresses for the next story. When new CGs are added to a story in a monitored rundown, the monitor program re-assigns all the addresses within that story to keep them contiguous. Using this option will reduce the number of times adding a new CG causes the address range to shift to a higher set of numbers.</p> |

Here is an example where all of the options are used:

```
casvr  casvr  bsvr  bcs-bcw
      video  airspace  -  3
      cg     deko1  deko  C:MSG  -          500  799
      cg     max    -     C:MAX  D:TEMPLATES  400  599  20
mossvr mosgw  -     event-mos
      mos    sony   -
```

In the above example, the information lines for two servers are shown before the device lines. The two servers are:

- a ControlAir Server called “casvr”
- a MOS Gateway Server called “mosgw”

Both a video device manager called “airspace” and a cg device manager called “deko1” are on the ControlAir Server, as well as a third device manger for a different character generator.

In the next example the first two device managers are individually configured on separate servers called “video” and “fonts”.

```
casvr  video  bsvr  bcs-bcw
      video  airspace  -  3
casvr  fonts  -  bcs-bcw
      cg     deko1  deko  C:MSG  -          500  799
```

5. Save the map story.
6. Test monitor the show to ensure map story entries are functioning correctly.

The monitor server only examines the show’s map story entry when a user turns it on, so any changes to the show’s map story entry will not take effect until then. Changes made to the show’s map story entry after the show is monitored do not take effect until the next time the show is monitored.

For this reason, you should monitor the show after creating or modifying the show’s SYSTEM.MAP story entry to test the changes you make. Monitoring the show allows the monitor server to check your work and ensures smooth operation when you produce the show.

Updating the iNEWS System Dictionaries

Machine Control System (MCS) dictionary files in iNEWS can be modified to customize the appearance of status indicators from various devices. For instance, an Omnibus device, connected to iNEWS, shows a video play-back status of “OnAir,” but the status field in the show rundown on iNEWS shows “Play” instead. If the system administrator wants the two status indicators to match, the MCS dictionary file in iNEWS must be modified. This will “translate” the iNEWS status wording so that it corresponds to what appears on the actual device.



The monitor server uses the MCS dictionary to show status for all devices, so ensure that words you choose apply to all devices. For instance, there is no way for the play status to be “OnAir” for character generators and “Playing” for video servers simultaneously.

Dictionary files are located in the /site/dict directory. This section covers changes made only to the /site/dict/mcs file; if changes are made to other dictionaries in the /site/dict directory, the procedure may require more steps.

In the following procedure, as an example, the term “Play” is changed to “OnAir” in the /site/dict/mcs dictionary file.

To edit the dictionary file:

1. Select all servers at the console, so changes you make are made to each server's copy of the file.
2. Open `/site/dict/mcs` for editing by typing: `ed /site/dict/mcs`
3. Navigate to the line with the word you want changed, such as "PLAY" by typing: `/PLAY`

The console will respond with a display similar to the following:

```
A_EVPLAY      /3PLAY
```

In the above example, PLAY appears twice in the line.



When navigating in the file, remember the line editor is case-sensitive. So, typing either `/Play` or `/play` will not locate a line with "PLAY."

4. Substitute the new word, such as "OnAir," for the second occurrence of the existing word, PLAY, by typing: `s/3PLAY/3OnAir`

The console will respond with a display similar to the following:

```
A_EVPLAY      /3OnAir
```

5. When you finish making changes to the dictionary file, save your changes and exit the line editor by typing (what appears in bold):

```
w
```

```
1279
```

```
q
```

Do not use an uppercase (W) in this step. When you press Enter, the editor responds by displaying a numerical value indicating the file size. See "[The Line Editor, ed](#)" on page 655 for more information.



See Appendix C for more information about Dictionary files.

If you change your existing MCS dictionary files, you must apply those changes by running the `makemontab` command at the `iNEWS` console; however, any monitor servers that are running at the time the command is entered will not apply the changes. The monitor servers must be stopped prior to running the `makemontab` command.



A system administrator who knows the device numbers for monitor servers can choose to stop just those programs; however, it should be done only during off-peak hours, when the monitor servers are not used to monitor on-air shows.

To update your iNEWS dictionaries:

1. Select all servers.
2. Enter superuser mode, using the current password.

3. At the specified time, select one server and type `list s` to check who is still logged in and which server programs are still running.

A message similar to the following appears:

```
S264    2Af4    B
R801    stevens A
```

The `list s` command displays the device controlling the session, the user account used for the session, and the server servicing the session. In the example, the first line starts with an S, indicating a server (utility program), such as a monitor server. The number after the S is the device number. If you are unsure whether the device number is that of a monitor server, check the iNEWS configuration file (`/site/config`) for monitor servers and their assigned device numbers. You can do this by using the `list c` command at the console, in the following format: `list c <device number>`.



Server programs are only listed as a result of the `list s` command if they are actually running, so they may be in use. Do not stop them without first checking with production staff to ensure you are not stopping a monitor server that is monitoring a show that is on-air.

4. After checking with production staff, type `list s` again to confirm whether all monitor servers are off.
5. Use the `stop` command followed by the device number(s) of monitor servers. This further ensures they are stopped. For instance, type: `stop 264`
6. Type `makemontab -i` to translate dictionaries used by monitor servers.
7. Log off the iNEWS console.

Unlike other utility programs, monitor servers do not have to be restarted from the iNEWS console. They are turned on and off from the iNEWS Workstation.

Creating Styles

Some sites use the monitor server alone—without any direct connection to devices, Command, or ControlAir—to check machine control events within scripts for errors and produce playlists that personnel can print out or refer to during broadcasts.

Whether the monitor server in iNEWS is used alone or with ControlAir to create a list of CG events, it must have a defined set of styles to work properly.

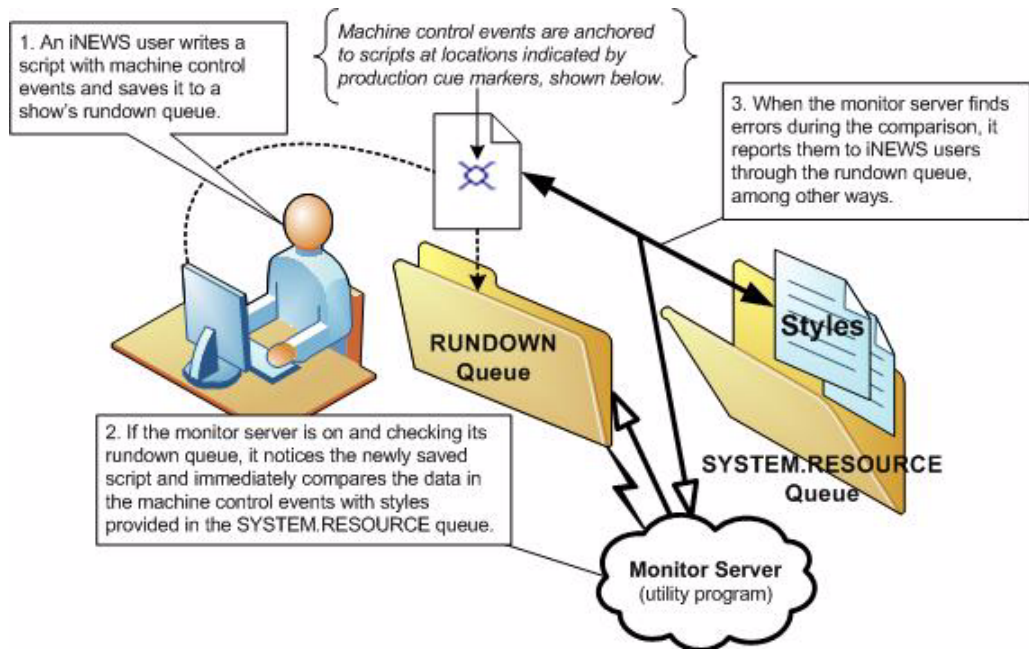
Styles apply to machine control events for character generators (CGs). Video servers and MOS devices do not use styles.

Styles are defined in stories located in the SYSTEM.RESOURCE queue of the iNEWS database. Each style represents elements necessary for the device (CG) to produce a particular playback event.

Each device must have its own list of styles. If there are two production devices of the same type, such as two CGs, users may assume that a style that works on one device also works on the other. Consider creating a new story for each device, defining the same styles for both devices to avoid confusion. Because SYSTEM.RESOURCE stories for several devices can be lengthy, users can find information they need more efficiently if there is one story per device, or one story per type/topic. For instance, one story for general lower-third CGs, and separate stories for sports, locations, etc.

Styles make it easier to create event requests because they simplify the information users must put into the request. Most importantly, styles provide the information required for the monitor server to check machine control events for errors, which is then reported to iNEWS users through instant messages or by changing the color of the event when displayed in the Story panel at an iNEWS Workstation. This information can also be forwarded to the ControlAir system.

The following diagram shows where styles appear in the workflow.



 *Step 2 in the diagram applies only to CG events.*

If you use effects in your character generator styles, you should develop a strategy for using them. How you use effects depends on whether or not you use the ControlAir system:

- If you have a ControlAir Workstation and want to include an effect in a style, you must first define the style effect in the DMP file for the device manager that controls the production device.
- If you do not have a ControlAir Workstation, the effects you include in your styles serve as instructions to your production device operators. Develop a standard set of effect names that your production device operators can easily understand.

Styles are stored in stories, which are saved to the SYSTEM.RESOURCE queue. These stories are created and edited like any other story in the iNEWS newsroom computer system. However, if the queue has a write security group assigned to it, you must belong to that group or be logged on as a superuser to edit stories in the SYSTEM.RESOURCE queue.



If your SYSTEM.RESOURCE queue does not have a write security group, one should be assigned. Because its stories define styles people will use to request playback events, assigning a write security group helps ensure that only authorized personnel can edit these stories. See “Groups” on page 151 for more information.

To create a style:

1. At an iNEWS Workstation, navigate to the SYSTEM.RESOURCE queue.
2. Do one of the following:
 - ▶ Open an existing story to edit.
 - ▶ Select File > New Story.

The new story is created and will appear with a blank Story Form (top) and Story Text panels, as shown:

| | |
|--|----------------------|
| <i>Title</i> | <i>Last Modified</i> |
| <input style="width: 95%;" type="text"/> | 9/11/2009 9:0 |

3. The story should begin with a commented header and parameter line similar to the following:


```
;Resource File
;Dev   Style   Template   Effect   # of Fields   ;Comments
```



Lines beginning with a semicolon (;) are comments and ignored by the system. Comments may also be added to the end of lines.

4. Do one of the following:
 - ▶ If editing styles in an existing story, move to the appropriate line(s) in the story.
 - ▶ If creating a new story, type the header as shown in the previous step.
5. Enter a style entry line for the device, ensuring all parameters are addressed, as explained in the following table.

| Parameter | Definition |
|-----------|---|
| Device | The device name of the character generator to which the style applies. This name is defined in the device’s entry in the ControlAir configuration file. For instance, CG, CG1, and CG2. |
| Style | The name of the style. Specify a name, eight characters or less, containing at least one alpha character. For instance, 1LINE , 2-LINE , and LLBOX are valid style names, but 412 is not. |

| Parameter | Definition (Continued) |
|-----------------|---|
| Template Preset | Enter the character generator address where the pre-recorded super or template that you want to use is located. |
| |  <i>Many CGs allow alphanumeric addresses. Template addresses for these machines can also be alpha-only addresses of 20 characters or less, which are very useful to CG operators. (Don't use the filename extensions.)</i> |
| Effect | <p>The name of a special effect you want to include in the style. If you do not want to include a special effect, such as FADE or WIPE, type a dash for this parameter.</p> <p>If you have a ControlAir Workstation and want to include an effect in a style, you must first define the style effect in the DMP file for the device manager that controls the production device.</p> <p>If the operator plays the event requests from the device's console, you can use any name up to eight characters long to represent the effect you want. However, ensure the operator is familiar with the name. When an event request uses this style, the monitor server puts the effect name you specify here in the event list entry for that event request. The operator uses the effect name to identify what effect should be used when the event is played.</p> |
| # of Fields | <p>If the style uses a template, use this parameter to specify the number of fields (or lines) in that template. Typing a number of fields here specifies to ControlAir the address refers to a template. The number here also ensures that this style cannot be used to add more text than there are fields in the template.</p> <p>If the style uses a pre-recorded super address, type a zero here to specify to ControlAir this parameter refers to a specific CG address. When you type a zero for this parameter, ControlAir ensures that this style cannot be used to add text to the CG.</p> |

6. When finished editing the styles in the story, save it by clicking the Save button or selecting File > Save.
7. Test new or edited styles by monitoring a rundown. Changes to SYSTEM.RESOURCE stories do not take effect until a show containing scripts with machine control events using the styles is monitored. Also, if a monitor server is running when changes are made, they will not take effect for that monitor server until it is restarted.

Some examples of styles for a character generator are shown in the sample SYSTEM.RESOURCE story below:

| Title | Last Modified | Created By |
|----------|---------------|------------|
| resource | 9/11/2009 9:0 | sysadmin |

| Dev | Style | Template | Effect | # lines | Comment |
|-----|---------------------|----------|--------|---------|-------------------------|
| cg | bnews | 62 | - | 0 | :breaking news |
| cg | mm | 75 | - | 1 | :money matters |
| cg | lottoph | 9993 | - | 12 | :lotto/powerball |
| cg | world | 50 | - | 1 | :the world |
| cg | nation | 50 | - | 1 | :the nation |
| cg | south | 50 | - | 1 | :the south |
| cg | recap | 52 | - | 1 | :recap |
| cg | nitetz | 53 | LOGO | 1 | :11 tonight |
| cg | tvex | 67 | WIPE | 0 | :tv exclusive |
| cg | intvl | 1 | - | 1 | :1 line interview |
| cg | intv | 2 | - | 2 | :2 line interview |
| cg | loc | 4 | - | 1 | :1 line locator |
| cg | locdate | 5 | - | 2 | :locator with date |
| cg | date | 6 | - | 1 | :date |
| cg | file | 7 | - | 0 | :file tape |
| cg | ctsy | 8 | - | 1 | :courtesy |
| cg | sktch | 10 | - | 1 | :sketches by |
| cg | locskylv | 24 | - | 1 | :live skycam locator |
| cg | repsky | 18 | - | 1 | :reporter skycam |
| cg | loclv | 19 | - | 1 | :live locator |
| cg | intvlv | 20 | - | 2 | :live interview 2 line |
| cg | live | 21 | - | 0 | :live |
| cg | replv | 22 | - | 1 | :live generic reporter |
| cg | lvsky | 23 | LOGO | 0 | :live skycam |
| cg | rep | 27 | - | 1 | :reporter not live |
| : | : | : | : | : | : |
| : | special for NoonDay | : | : | : | : |
| cg | locn | 1002 | - | 1 | :1-line locator/noonday |
| cg | innd | 1001 | - | 2 | :2 line interview noon |
| cg | lond | 1008 | - | 1 | :live locator 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 |

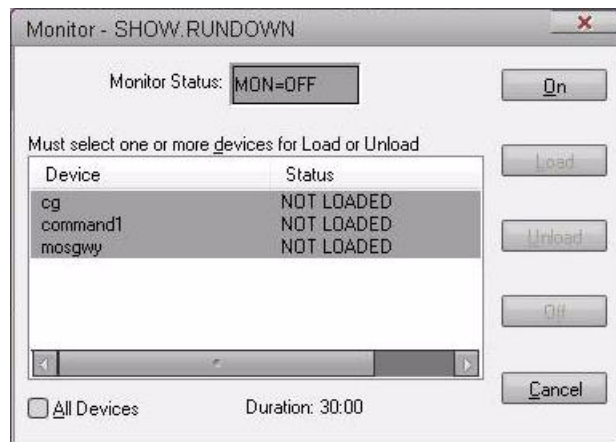
Using the Monitor Server

This section provides some basic user information, which can be used to test when the monitor server is set up and working properly. Procedures include how to turn the monitor server on and off, as well as how to instruct it to load or unload machine control event data.

To use the monitor server:

1. Log in to an iNEWS Workstation.
2. Open the appropriate rundown queue.
3. Select Tools > Monitor.

The Monitor Queue dialog box appears.



The automatic shut-off time appears at the bottom of the dialog box as either a Duration (shown) or Quit Time.



Duration indicates how long machine control will remain on once the monitor server is turned on. Quit Time indicates a specific time (in 24-hour format) that machine control is automatically turned off.

4. Select a Monitor Mode based on the following options:

| Option | Description |
|--------|---|
| On | Turns the monitor server on and creates an event list for each device in the rundown queue. |
| Load | Instructs the monitor server to load event lists to devices you select. |

| Option | Description |
|--------|---|
| Unload | Clears device event lists (also known as playlists) without turning the monitor server off. |
| Off | Turns the monitor server off, overriding the Quit Time setting. |



The status bar in the iNEWS Workstation displays the current status of the monitor server, such as: MON=OFF, MON=ON, MON=LOAD, and MON=PART, which stands for partial device loads.

The Devices section of the Monitor Queue dialog box displays a list of devices based on those defined for the rundown queue in the SYSTEM.MAP story.

The list could include any one or all of the following servers, such as:

- Command Server
- ControlAir Server
- MOS Gateway Server

If you select Load, specify the device or server to load from the Devices list, or select the All Devices checkbox to load all devices associated with the rundown queue. To select multiple devices from the list, without selecting all, press the Ctrl key while you click on each item. If you select a server, the data will be loaded to all devices associated with that server.

If you select the MOS Gateway Server that is interfacing with a device that has multiple channels, such as an OmniBus automation system, a second dialog box will appear, allowing you to select the channel from a drop-down list. To enable this workflow, the MOS server name and MOS device name must match.



5. Click OK to return to the previous dialog box.
6. Click OK again.

Network iNEWS Systems Using RX/TX Links

This section introduces you to rx/tx links and explains their operation. Use rx/tx links to transfer stories between iNEWS systems and other systems using the Ethernet network. To send stories across a network connection, use network rx/tx links. You can connect several servers or systems into a network using rx/tx links.

For one system to send stories to another, it must have a Tx (transmit) link configured. The other system must have an Rx (receive) link configured to receive stories the tx link sends.



Information relevant to this section is also contained in “Field Validation” on page 335.

Sending Story Forms

Your system can transmit story form information by sending the form name with a story or by sending the full form with a story. The first way reduces the time it takes the system to send a story to another system by omitting the form and sending only the text of the story, including any text contained in the form’s fields, and the form name. When the receiving system gets a story without a full form, it uses the form name supplied with the story. The form may not exist on the receiving system, but no check is made by the rx link.

If the receiving system gives a story a new form, it fills in the fields of that form as much as possible with the story’s original field text. For transfers using the NSML format, if the original story had fields that are not present in the new form, the text from those fields are stored in the field section of the story, but are not displayed on screen and will be discarded when the story is modified.

To ensure that transmitted stories have the same form on both systems:

- ▶ Add a sendform line to the Tx job list.

This instructs the system to send the full form with the transmitted stories. The sendform line must appear in the job list immediately following the scan line.



For transfers using the NSML format, if using the tx sendform option, the queue on the receiving system should have the FORMS_ALLOWED attribute. Avid recommends that if the volume of stories transferred is high, whatever forms are used for stories should be resident on both systems, especially when the two systems are owned by the same customer.

Setting Automatic Update

When a new version of a story is transmitted to a queue that already contains an older version of the same story, you usually want the receiving system to replace the older version of the story with the new one. To do this, enable the update trait on the queue where the received stories are placed.

For instance, suppose an affiliate in Washington sends stories it uses for its six o'clock show to your system as these stories are added to or modified in the show's rundown queue. On receipt of these stories from Washington, your system puts them in the queue called WASHINGTON.RUNDOWN.6PM. You want your system to dispose of old versions of stories as new versions come in from Washington.

To set an automatic update:

1. Log in as a system administrator at an iNEWS Workstation.
2. Navigate to the queue, such as WASHINGTON.RUNDOWN.6PM, and right-click on it.
3. Select Properties. The Queue Properties dialog box appears.
4. Click on the Maintain tab.
5. Click on the Update check box to turn on the update database trait in the queue.
6. Save changes and close the dialog box.



If you change the update trait and stories are in the queue, those stories will not be replaced by updated versions. Turn on the trait when the queue is empty.

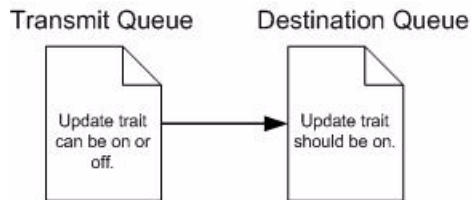
After you apply the update database trait, your system starts replacing older versions of stories in WASHINGTON.RUNDOWN.6PM with new ones as they come in from Washington.

If you have a tx link that sends stories to an update queue, any change you make to a story in a queue watched by the tx link affects the copy of the story in the other system. This includes killing a story, which causes the tx link to notify the receiving system that the story has been killed. The second system then kills its copy of the story as well. You can use the ignore-del command to prevent the tx link from notifying the receiving system when stories are killed.

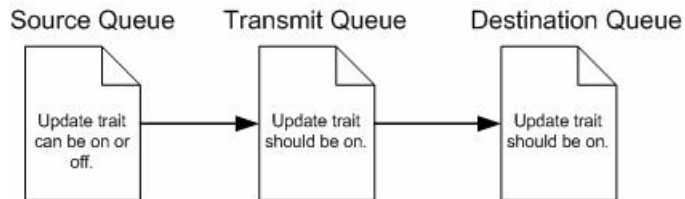
Update Trait - Queue Considerations

A story's final destination queue and any intermediate queues through which the story passes before reaching the destination queue must have their update trait turned on. If not, the final destination queue does not get updated. It does not matter if the queue where the story was created has its update trait turned on.

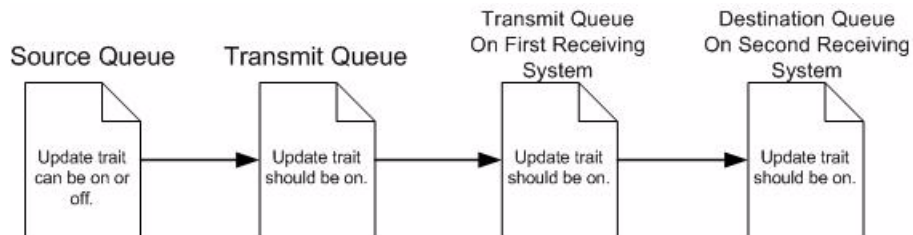
The example below shows which queues must have their update trait turned on if a story is transmitted directly from the queue in which it is created to its final destination queue.



If the story passes through an intermediate queue before reaching its destination queue, both the intermediate queue and destination queue must have their update traits turned on, as shown below:



Similarly, if a story received from one system is copied to another queue for transmission to a third system, enable the update trait on every queue through which the story passes, as shown below:



Changing Queue Order

Ordinarily, when a queue in the transmitting system is ordered, changes are not evident in the receiving system's corresponding queue. This is adequate for many queues, because their story order is not important. However, story order is very important for other queues (for example, rundown queues). These queues must be identical on the local and remote systems.

To have the system transmit queue ordering information:

- ▶ Add an order line to your network tx job list.

When a local queue is ordered, information is sent to the remote queue. That queue is ordered, identical to the local queue. This also applies to action servers.



The destination (remote) queue must have its update trait turned on.

Also, additions and deletions to a sorted queue on the local system do not cause the system to transmit queue order change information. Ensure that you turn on the sort attribute in the destination queue.

Adding Rx/Tx Links

If your iNEWS system is connected to the same network as another iNEWS system, you can use network rx/tx links to allow one system to automatically send stories to the other. For each rx or tx link, create an entry in your system's configuration file that defines where the link runs and how it sets up.

You can also use network rx/tx links to have a networked iNEWS system periodically send stories ready for archival to a third-party archive system on the same network. Set these links up so that people can place stories ready for archival in archive queues on the iNEWS system. Then, on Sunday morning, the iNEWS system moves these stories to queues on the archive system.

Give the iNEWS system a tx link that performs two tasks. At 3 a.m. every Sunday, the first task moves any stories added to ARCHIVE.SCRIPTS since the previous Sunday to SYSTEM.TRANSFER.SCRIPTS on the archive system. Then, at 4 a.m. every Sunday morning, the second task moves any stories added to ARCHIVE.PACKAGES that week to FROM.NEWS.PACKAGES on the archive system.

To add network rx and tx links to your iNEWS system:

1. Ensure both systems are in the hosts file or can be resolved through DNS.

For two systems to use network rx/tx links to exchange stories, each must be network resolvable, via either /etc/hosts or DNS. To test DNS, type `ping -c 3 <server name>` on the console command line for each iNEWS Server.

Use the `cat` command to check each system's hosts file and ensure that necessary servers are listed. For instance, check the iNEWS system:

```
NRCS-A$ cat /etc/hosts
```

A message similar to the following will appear:

```
125.0.0.1    NRCS-A
125.0.0.2    NRCS-B
125.0.0.3    ARCHIVE-C
...
```

The archive system's server (ARCHIVE-C) is listed in this file. Now check the archive system's hosts file for NRCS-A and NRCS-B.

If you find that either system's hosts file does not contain the names of the computers belonging to the system to which you want it to connect, call Avid for help in modifying the file.

2. Choose device numbers for the links.

Reserve a range of numbers for network devices and always choose the next available number from within that range. In the examples for this procedure, we have reserved numbers 301-399 for network devices.

To see which numbers are taken, check your system's configuration file. For this example, the device number 301 is chosen for the network tx link. On the archive system, we use device number 101 for our network rx link.

3. Create the tx link job list queue in the database.

See [“Creating a New Queue” on page 104](#) for more information.

This queue must be in SYSTEM.CONFIGURE. Its name must begin with a three-digit number matching the tx link device number.

For instance, create a queue named 301 to hold the job list. Follow the number with a hyphen and a descriptive name, such as: `-txnet-to-archive`. The full pathname of the queue would be `SYSTEM.CONFIGURE.301-txnet-to-archive`.



Network rx links do not require job lists.

4. Open the newly created tx link job list queue.

5. Create the tx link job list story in the database.

Like action servers, tx links use mailbox and timed-interval tasks. Tasks that you want the tx link to execute as soon as a story is modified in a queue must be mailbox tasks, while those that you want the tx link to execute at a specific time must be timed-interval tasks. See [“Types of Tasks for Servers” on page 320](#) for more information.

In a network tx link's job list, you must always follow the scan line with an open line and a put line, both of which are used only in this type of job list.

The open line uses the open command in the following format to open a network connection to the receiving system:

```
open <computer name>[:<port>] <username> [<format>]
```

The computer name must match be network resolvable. Use the ping command to test.

The port is the port number used by the rx link on the receiving system. If it is omitted, the tx link program will use the port defined by the rxnet service in its local /etc/services file; if there is not an rxnet service defined, the tx link program will look up the FTP service in its local /etc/services file.

The user name must have the same password on both systems.



If going to a third-party system, be aware that iNEWS sends all passwords in lower-case.

To improve security of your tx link, apply the blacklist user trait to the user name (account) you list here on both systems, which prevents anyone from logging in under that name. This has no effect on name use in a job list. See “User Traits” on page 69 for more information.

If the format is HTML, then the open command syntax has two more parameters. See “Web Publishing and Access” on page 429 for more information.

Possible formats are:

- NSML - News Story Markup Language
- 2NSML - 2nd version of NSML contains extensions for MOS support. Default format.
- HTML - Hypertext Markup Language
- SEP - Avid Netstation Story Exchange Protocol



The rxnet program uses an inactivity timer set to the system localtimeout value. If rxnet does not receive a command within the inactivity interval, it will exit. If the localtimeout value is zero (0), the inactivity timer is disabled. You can and should set a separate inactivity timer for rxnets by creating the story /site/env/rxnet and using the RXSITEDILE=<# of seconds> variable to control rxnet timeouts.

You must follow the open line with a put line, which uses the put command in the following format to send stories to a queue on the receiving system:

```
put <target queue name>
```



The target queue name may be formatted using the strftime syntax, may be whatever directory pathname that is appropriate for the target system if not an iNEWS system, and will be automatically created if it doesn't exist, and if the target system allows.

If you do not name a queue, the put command uses the destination queue assigned to the user you specified in the open command.

If you want stories you transmit to be distributed to more than one queue, set up an action or distribution server on the receiving system to do this. See [“Adding an Action Server” on page 333](#) and [“Adding a Distribution Server” on page 350](#) for more information.

In this example, the network tx link's job list requires two timed-interval tasks. At 3 a.m. each Sunday, the first task sends copies of everything added to ARCHIVE.SCRIPTS in the last week to SYSTEM.TRANSFER.SCRIPTS on the archive system. Then, at 4 a.m. every Sunday the second task sends copies of everything added to or changed in ARCHIVE.PACKAGES to FROM.NEWS.PACKAGES on the Archive system.



To ensure forms are transmitted with stories, include a sendform line in the Tx job list. The sendform line must be added to the job list anywhere after the scan command and applies to all following put commands. The sendform option setting persists to the end of the current task or until a sendform no command changes the setting within that task.

If you want any queue order changes in your scan queue to be transmitted to your remote queue, include an order line in the Tx job list. The order command must precede the open command to affect the put command. Also, make sure the update trait of the destination queue is turned on.

After the tx link has completed the first tasks, you may want it to remove stories it sent from the scan queue. To have it do this, end both tasks with the remove command.



If you use either move or remove, it must be the last command in the task's command set.

The finished job list looks like the following:

```

SLUG                                CREAT
307 Job List                        3/16/2
at 3:00
on sun
  scan archive.scripts
  open archive link|
  sendform
  put system.transfer.scripts
  remove

at 4:00
on sun
  scan archive.packages
  open archive link
  sendform
  put from.news.packages
  remove

```

You can use the following job list commands in a network tx link job list. (These are the same commands used by action servers.)

| | | |
|---------|------------|----------|
| at | ignore-del | remove |
| bscan | mailto | replace |
| dup | move | scan |
| every | number | send-del |
| fast no | on | sendform |
| ignore | order | validate |
| | quiet | |

The following commands can only be used by tx link: open, put, publish, and extension.

The mailto joblist command's syntax is:

```
mailto <recipient1> [<recipient2> ...]
```

The list of recipients is a space separated list; do not use commas.

The effect of this command is to send the modified story from the scan queue as an email text message.

The sendform joblist option causes the field contents of the modified story to be prefixed to the story's text.

The story form assigned to the modified story is used to determine which fields are included and the order of the fields. When placing a field into the email text message, the field will be preceded by the label text from the story form. Each label/field pair will be put on a separate line of the email text message. If the field does not have any label text, the field name is used.

The AIR-DATE, MODIFY-DATE, and CREATE-DATE fields are formatted with the strftime format: %x %X. This format can be controlled by setting the environment variable—VT_TIME—to the desired strftime format and including it in the action / txnet program environment. Contact Avid for assistance.



Only the modified story's text (and field text if the sendform option is "on") is put into the email text message. No production cues, anchor points, nor attachment text is put into the email text message.

If write access for the destination queue on the receiving system in the put command is restricted, the connection must be established with the appropriate security to send stories to this queue. The user name you specify in the task's open command must belong to the necessary group on the receiving system.



Usually, you assign a read group to the System directory that prevents most users from reading stories in that directory. If you do not want to restrict all of the System directory, consider restricting who can modify stories in SYSTEM.CONFIGURE.

6. Add links to the configuration file on each iNEWS Server in your system.

- a. Select all servers.
- b. Open and edit the configuration file, by typing (what appears in bold):

```
NRCS-A$ ed /site/config
1259
```

After you press Enter, the editor responds by displaying a numerical value indicating the file size expressed as the number of characters, including spaces and returns.



This procedure, which modifies the /site/config file uses ed, the line editor. If you do not know how to use ed to modify lines in the file, please see "The Line Editor, ed" on page 655.

- c. Add the network tx link's device number to a reslist line in the appropriate host definitions. For instance, put the network tx link on server A by adding a servers line in its host definition beneath the servers line, which is the last line in the host configuration.

```
servers 256 257 258 259 260 261 262 263
servers 301
```



To keep the network tx link running if one of the iNEWS Servers fails, add this line to each alternate host definition before proceeding.

7. Add a server line for the tx link, using the format shown:

```
server <device #> txnet <mailbox> <device name>
```

The finished line (in bold) for the tx link should appear at the end of the line grouping:

```
server 256 action          256 actphon ;action svr
server 257 distribution    257 devname1 ;dist server
server 258 parallel       258 devname2
server 259 keyword        259 key1 ;keyword server
server 260 seek           260 seek ;seek server
server 261 ftsseek        261 - ;fts searches
server 262 ftsindex       262 - ;fts indexing
server 301 txnet          301 - ;tx link
```

8. Create a server line for the rx link. Like the server line for the tx link, place this line at the end of the file.

The general format for a network rx link configuration line is similar to that of a network tx link:

```
server <device #> rxnet [<IP address> or <MAC address>] <device name>
```

The rx link does not use the device name field so place a hyphen in that position. If an IP address is specified for the fourth column, the rx link will only be used for connections from the system with that IP address.

The rx link and tx link would be on different systems if supporting a job list like the one shown below:


```

SLUG                                CREAT
307 Job List                         3/16/2
at 3:00
on sun
  scan archive.scripts
  open archive link|
  sendform
  put system.transfer.scripts
  remove

at 4:00
on sun
  scan archive.packages
  open archive link
  sendform
  put from.news.packages
  remove

```

The finished line should look like this:

```
server 101 rxnet - -
```

- When you finish making changes to the configuration file, save your changes and exit the line editor by typing (what appears in bold):

w

1279

q

Do not use an uppercase (W) in this step. When you press Enter, the editor responds by displaying a numerical value indicating the file size. See [“The Line Editor, ed” on page 655](#) for more information.

- Test your configuration changes. See [“Testing the Site Configuration File After Alteration” on page 253](#) for more information.
- Reconfigure the system. See [“Incorporating Configuration Changes” on page 253](#) for more information.

Repeat this step on the archive system to incorporate the network rx link into its operation.

- Start the tx link using the following command format: `restart <device #>`

For instance: NRCS-A\$ **restart 256**

When the system displays a Hot-to-go message, the tx link has been successfully started.



A network rx link is not restarted after you add it to your system. The system starts the rx link automatically as soon as it receives something from the transmitting system.

13. (Optional) Back up site files using sitedump command.

14 iNEWS Community

The iNEWS Community feature that allows customers with multiple iNEWS systems to share content and collaborate on stories. A user can view, edit, and search for content stored on any of the iNEWS systems from a single iNEWS Workstation. Community is a licensed feature, however, and requires some configuration to enable it. This chapter provides the details for how to do that.

This chapter contains the following main sections:

- [Configuring iNEWS for Community](#)
 - [Syntax of the ctraits Command](#)
- [Viewing Remote Systems or Community Sessions](#)
 - [Restrictions](#)
 - [Local and Remote SYSTEM.MOS-MAP](#)
- [Removing a System from Community](#)
- [Connection Issues](#)
 - [System Messages in the iNEWS.log](#)

Configuring iNEWS for Community

In this section, the sample iNEWS systems are two dual-server systems identified as: WAVD-A/B and WNWS-A/B.



If upgrading from a version earlier than iNEWS v2.6.5, you need to create the Community database file by using the `dbgen c` command at the console prior to doing the following configuration procedure.

To configure iNEWS for Community:

1. Ensure all servers can ping each other by name, and that all workstations can ping all servers by name.
2. Ensure all systems have unique site keys. Use the `status 1` command at the console to display the site key.
3. Modify `/site/config` on each system accordingly:

- a. Remove any Remote Search devices. Remote Search has been replaced by Community.

- b. Create devices for the Community connections, namely `cinws`, as shown below:

```
host ab a
reslist 801 803 805 807 809 ; Community
host ab b
reslist 802 804 806 808 810 ; Community
; INWS (COM and RSEARCH) SESSIONS
cinws 801:810 - gnews - ; Community
```

4. (Optional) To activate the Community logging feature, you must modify the `W_LOGTYPES` token in `/site/dict/words`, as shown:

Before: `W_LOGTYPES=/D`

After: `W_LOGTYPES=/DC`

5. (Optional) If you would like to restrict the remote iNEWS Server versions that can connect to the local system, create the queue `SYSTEM.CLIENT.COMMUNITY-VERSIONS`, and insert a story with the allowed server versions listed.

6. Create generic users with usernames matching the incoming systems' names. The permissions set for these users' traits define the permissions of generic incoming connections from servers of those same names.



If the generic user for the local system does not exist on the remote system, users on the local systems whose usernames do not exist on the remote system will not be able to connect to the remote system.

- On WAVD, create a user name that matches the connecting system name, such as WNWS.
 - On WNWS, create a user name that matches the connecting system name, such as WAVD.
 - Local users who are not explicitly defined on the remote servers will log in to the remote servers with the properties defined by these generic users.
7. If you would like to restrict which non-superuser users can see other systems, create groups defining who can connect to the target systems. For example:
 - On WNWS, create a group name, such as grpWAVD. The group name has no requirements; grpWAVD is just an example.
 - Add all non-superuser users who should be able to connect remotely to WAVD to the group.
 8. Add each system to the other system's Community using the ctraits command in the following format: `ctraits <system name> [<option> <value>]...[+flag]...[-flag]...`
 - For instance, on the WAVD iNEWS Server, in superuser mode, enter the following command: `ctraits WNWS suffix AB +m +o +i`
 This command will add the system WNWS to the local system's Community file, instructing the local system to connect to the server WNWS-A and check for the requested username. If WNWS-A is unavailable, connection to WNWS-B will be attempted. If the suffix were reversed to BA, then connection to the B server would be attempted first.
 A system cannot be added to its own Community file.



A suffix or list of suffixes is required, and the +m flag is required to allow messaging to the system. For more information, see “Syntax of the ctraits Command” on page 422.

- On the WNWS iNEWS Server, in superuser mode, enter the following command:
`ctraits WAVD suffix AB group grpWAVD +m +o +i`
 In the above example, only superusers and users in the grpWAVD group will see and be allowed to connect to WAVD from WNWS.



Changes made do not take effect until the next time the systems are configured.

9. Take each system offline by typing: **offline**
10. Configure each system by typing: **configure**
11. Bring each system back online by typing: **online**

After you have configured both systems and they are back online, the systems should be ready to go.

Syntax of the ctraits Command

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

The syntax for the ctraits command is:

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

The options and flags are explained in the following table:

| 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. |

Viewing Remote Systems or Community Sessions

You can use variations of the list command to view information about remote systems and Community sessions.

To view the remote system in the local system's Community:

- ▶ Enter `list C` (with a capital C) on the iNEWS Server, which will display the following:

```
SYSTEM SUFFIX MOI GROUP
WAVD AB moi grpWAVD
```

For more information, see [“list C” on page 503](#).



Permissions on the remote system are defined by usernames on the remote system. A user's local permissions are disregarded by the remote system.

To view Community sessions at the console:

- ▶ Enter `list s` on the iNEWS Server.

The system will display a list of current sessions. Community sessions appear in the output as C sessions.

```
G850 jsmith A
C950 tjones A
C951 jdoe A (c)
```

The G session for user jsmith is an example of a standard local iNEWS session (inws). The C session for user tjones is a Community session (cinws) using the person's true username. And the C session for user jdoe is a Community session (cinws) using the generic username. This means the jdoe user name does not exist on the remove server.



The localonly user trait, when applied, restricts remote users with that username from connecting through Community. For example, setting the generic “avid” superuser to localonly will cause incoming users named “avid” to be rejected at log in. The localonly user trait can be applied by either selecting the Local Only checkbox in the User Properties dialog box or by using the +localonly option with the utraits command at the console.

Restrictions

Users will experience some restrictions when connecting via Community.

- Local users cannot modify queue, directory, or user properties on remote systems.
- Users connecting as generic users will be prohibited from using EasyLock (also known as UserLock) to lock stories and directories; however, KeyLock is available.

Local and Remote SYSTEM.MOS-MAP

Stories and items are validated against the destination system's SYSTEM.MOS-MAP.

Plug-ins that create MOS or CAP events will create events that display the AMCPID appropriate for the destination system.

When stories containing events are moved to other systems, the destination system will attempt to match MOSID to AMCPID using the destination SYSTEM.MOS-MAP.

For Example:

- In SYSTEM.MOS-MAP, WAVD has: MOS001 WAVDamcp
- In SYSTEM.MOS-MAP, WNWS has: MOS001 WNWSamcp

In this example, when a story containing a MOS001 event is moved from one server to another, the AMCPID defined in the destination server will be used when the story is saved.

If there is no matching MOSID on the destination server, no modification is made to the event, and the following system message appears:

```
WNWS-B C908: Community move: mosID [MOS001] not in MOS-MAP
```

Removing a System from Community

You must be in superuser mode at the local system's console to remove another system from its Community. In the following procedure, the dual-server system WAVD A/B is used as an example.

To remove a system:

1. At the console, in superuser mode, enter `ctraits WAVD remove`.
2. Take the system offline by typing: `offline`
3. Configure the system by typing: `configure`
4. Bring the system back online by typing: `online`



Changes made do not take effect until the next time the systems are configured.

Connection Issues

If experiencing connection issues, here are some troubleshooting ideas to consider:

- Ensure all systems can ping other community members by name.
- If DNS issues appear, add `/etc/hosts` definitions.
- If a node is available but offline, the entire system is considered offline. The next suffix will not be attempted.
- An unreachable system will introduce a delay in message name checking. The server attempting to connect will wait for a response, but will force a timeout after five seconds.
- If a user cannot connect, ensure that either their username exists on the remote server or that the remote server has a user with the local system's name. If neither exists, the connection will be refused. If one of the names does exist and the connection is still refused, ensure that the username does not have the `localonly` trait set on the remote server.
- If connection is refused, ensure that the remote system has the local system listed as `+i`.
- If connection is refused and no system messages appear, ensure that the remote system has available `cinws` sessions.
- If connection is refused, ensure that the connecting server is of an appropriate version, as defined in `SYSTEM.CLIENT.COMMUNITY-VERSIONS`.

System Messages in the `iNEWS.log`

The systems, both local and remote, log connection refusal reasons in a log file, which should be consulted whenever connection issues arise. The log is located at `/var/log/iNEWS/iNEWS.log`.

In the following examples, the local system is `WAVD`, and the remote system is `WNWS`.

- Remote system does not have local system as an allowed incoming connection (`ctraits wnws suffix ba +m +o -i`)
 - Local message: `WAVD-A G520: Not authorized to connect to WNWS-B`
 - Remote message: `WNWS-B work: 172.24.96.48 is not in Community`
- Remote system does not allow local system's version in `SYSTEM.CLIENT.COMMUNITY-VERSIONS`.
 - Local message (appears once for each suffix attempted):
 - `WAVD-A G521: No connection to WNWS-B (not compatible)`
 - `WAVD-A G521: No connection to WNWS-A (not compatible)`

- Remote message: none
- Local system cannot find remote system in DNS or /etc/hosts.
 - Local message: At login (appears once for each suffix attempted):
WAVD-A G521: Unable to get messages from WNWS-B - Unknown host
WAVD-A G521: Unable to get messages from WNWS-A - Unknown host
 - Local message: At connection attempt (appears once for each suffix attempted):
WAVD-A G521: Unable to connect to WNWS-B - Unknown host
WAVD-A G521: Unable to connect to WNWS-A - Unknown host
 - Local message: At message send (appears twice for each suffix attempted):
WAVD-A G521: Community send to WNWS-B failed 521:unknown host
WAVD-A G521: Community send to WNWS-A failed 521:unknown host
 - Remote message: none
- Local system has an IP address for remote, but remote is not running.
 - Local message: At connection attempt (appears once for each suffix attempted):
WAVD-A G519: Unable to connect to WNWS-B - Timed out
WAVD-A G519: Unable to connect to WNWS-A - Timed out
 - Local message: At message send (appears twice for each suffix attempted):
WAVD-A G519: Community send to WNWS -B failed 422:connection
timed out
WAVD-A G519: Community send to WNWS -A failed 422:connection
timed out
 - Remote message: none
- Remote system is offline.



If the remote system's primary suffix is offline, the entire system is considered offline.

- Local message: WAVD-A G521: Community server WNWS-B is offline
- Remote message: When local user logs in to local system:
WNWS-B work: System is currently OFFLINE (30:172.24.96.48)
- Remote message: When local user attempts connection to remote system:
WNWS-B work: System is currently OFFLINE (29:172.24.96.48)
- Remote system does not have any 'cinws' sessions configured.
 - Local message: WAVD-A G522: No configured sessions on WNWS-B
 - Remote message: none

- Remote system is out of available 'cinws' sessions.
 - In this situation, there will be no system messages printed. The client will show WNWS is unavailable.
- Remote system is missing both the local username and the local generic name.
 - Local message: WAVD-A G519: Cannot login to community server WNWS (32)
 - Remote message:
WNWS-B C906: [4453] gnews 906 C-WAVD - 2.7.0.17 Hot-to-go
WNWS-B C906: Attempted login wavd 172.24.96.48 2.7.0.17 from WAVD
- Destination server does not have a matching MOSID for an event moved from the source server:
 - Source server message: none
 - Destination server message: WNWS-B C908: Community move: mosID [MOS001] not in MOS-MAP

15 Web Publishing and Access

The iNEWS Web publishing feature allows users to publish news on the World Wide Web (WWW) by transmitting stories to other computers in Hypertext Markup Language (HTML) format. Web Access allows direct, read-only browsing of the iNEWS database through a Web browser. This chapter covers the Web publishing installation and process, as well as Web Access.

This chapter contains the following main sections:

- [Web Publishing](#)
 - [Setting up Txnet to Send HTML](#)
 - [The HTML Export Template](#)
 - [Publishing iNEWS Stories to the Web](#)
- [Web Access](#)
 - [The Web Server](#)

Web Publishing

Configuring Web publishing for iNEWS involves the following:

- Setting up Txnet to send Hypertext Markup Language (HTML)
- Modifying the default or creating a new HTML export template and putting it into a queue

After the setup is complete, you can complete the Web publishing process. See [“Publishing iNEWS Stories to the Web” on page 446](#) for more information.

Setting up Txnet to Send HTML

To set up txnet to send HTML:

1. Establish Tx links.

See [“Network iNEWS Systems Using RX/TX Links” on page 407](#) for more information.

The links (txnet) transfer stories between systems. You can send stories to non-iNEWS systems. The only requirement is that the receiving computer supports File Transfer Protocol (FTP) so the iNEWS Server can send stories to it.

2. Get the name of the Web server to which you will send the stories and be sure it is listed in /etc/hosts.
3. Create a valid user account on the Web server.
4. Create that same user account on the iNEWS Server.



The Web server and iNEWS Server must have matching logins with matching passwords.

5. Create an entry in the job list for the txnet to specify sending stories in HTML format rather than the standard iNEWS format, News Story Markup Language (NSML).

open <computer name> <user name> HTML [<queue name>] [<story name>]

- a. Specify the Web publishing server name as the computer name.



The computer name can include a suffix in the form of :<port>. If :<port> is omitted, the txnet program will use the port defined for the rxnet service in the /etc/services file and if there is not an rxnet service defined, the txnet program will look up the FTP service in the /etc/services file. This allows the txnet program to connect to computers that use a different port for FTP connections.

- b. Specify the user name that gives you access to the Web publishing server (common user name and password).
- c. Specify the format as HTML to enable Web publishing.
- d. (Optional) Specify the <queue name> as the queue that contains the HTML export template. If no queue is specified, a default is used. See [“The HTML Export Template” on page 432](#) for more information.



If you specify a queue, the entire queue name must be specified in the open command. Otherwise, the system will issue a “Failed to get form from...” error message when txnet is restarted. Also, if you specify a queue that does not contain an HTML export template, Web publishing will not work.

- e. (Optional) Rather than use the default, which is the first story in the queue, specify the story name in the queue.

Here is an example of a correctly written open command line:

```
open ntsrvr webtx html system.webforms.s.script-tx
```

6. After you have edited the txnet job list, restart txnet for your changes to take effect.

When HTML stories are placed on the Web server, they are assigned a unique filename, similar to the following:

```
f6daab8.html
```

```
f6daab9.html
```

```
f6daaba.html
```

Additionally, a story named published.html can be created in the directory written to. The published.html file contains a listing of each file exported to the server, in that directory, referenced by the quick index field of the story. The format is:

```
<A HREF="qstamp.html">quickindex &nbsp;   </A><BR>
```

Here is an example:

```
<A HREF="f6daab8.html">pre-show tease &nbsp;   </A><BR>
```

```
<A HREF="f6daab9.html">bosnia &nbsp;   </A><BR>
```

```
<A HREF="f6daaba.html">fort bragg &nbsp;   </A><BR>
```

This story could be used as a base for creating an index page. Generation of the published.html file can be turned off by a job list command after the scan line: publish no. The default for this job list command is yes. See [“Job List Commands” on page 527](#) for more information.

A txnet job list command, called extension, is used to control the suffix for file names when sending stories from txnet using the HTML format option. The syntax is:

```
extension <suffix>
```

The `<suffix>` is any string of characters. When the story is output, the story name will be constructed as:

```
<qstamp as 8 hexadecimal digits> . <suffix>
```

The `<suffix>` does not include the dot and will also apply to the published file. So, instead of that file always being called `published.html`, it will now be called `published.<suffix>`. If no extension command is found in the job list, the suffix will be set to `html`.



It is also possible to use this same process to export data formatted as XML rather than HTML.

The HTML Export Template

The HTML export template defines the appearance of stories transmitted out of iNEWS—that is, exported in HTML format. The template is stored as a news database story whose text is in HTML, with optional embedded references to story entities.

The iNEWS system contains an HTML export template to use as the default, but you can customize it or create new templates. The default HTML export template is shown in [“Default Story Template” on page 449](#).

If you do not specify a queue in the job list where the HTML export template resides, the system uses the default location. The default location is constructed from the dictionary entries `Q_WEBPUB_FORMS` and `W_WEBPUB_FORM`. These entries are normally defined to be `SYSTEM.WEBFORMS` and `PUBLISH-FORM` respectively; therefore, the default location is `SYSTEM.WEBFORMS.P.PUBLISH-FORM`. If this queue does not exist, or does not have an HTML export template, a hard-coded default Web form is used instead.

Story Entity References

In creating the HTML export template, you use HTML tags and add story components or entities.

Specify entities in the HTML export template using an ampersand (&) to mark the beginning of an entity and a semicolon (;) to mark the end. When put in this notation, an entity is called a story entity reference.

The following table shows the three basic entities of a story, the format for each basic story entity, and its corresponding story entity reference:

| Basic Entity | Definition and Format |
|--------------------|---|
| Field IDs | <p>Story information such as story title, creation date, writer, and so forth. It is the information you see in a Queue or Story Form panel.</p> <p>Field ID format is: <code>&f-<i><field_id></i>;<i>[<optional format string></i>]</code></p> <p>The field ID is replaced in the HTML output by information associated with it. For instance, to create a field ID such as the title of a story to appear in the HTML template, use the following: <code><title>&f-title;</title></code></p> <p>You are combining HTML tags with the story entity reference. The information in the field ID is now a title in the HTML export template, and when it is merged with an actual story, it will contain the story's title, in HTML format. For instance, the title will look like this:</p> <pre><title>LOTTERY-WIN</title></pre> <p>The optional format string defines the format to be applied to the field ID entity. The string applies to fields that are time-related, such as the date or length of a story. Format strings are optional and defaults are assumed if they are not specified. See the section on optional format strings for more information on how to modify this default display for date or time fields.</p> |
| Body | <p>Story text.</p> <p>Body format is: <code>&body;</code></p> <p>When encountered in the export template, it is replaced by the story text in the resulting HTML output.</p> |
| Production cue set | <p>Production cues entered in the Instruction pane—to the left of a story's text.</p> <p>Production cue format is: <code>&aeset;</code></p> <p>When encountered in the export template, it is replaced in the resulting HTML output by a list of production cues defined in the production cue set, a collection of all production cues for the story. Production cues—also known as anchored elements—are anchored at specific positions in the story text. How they display depends on their content and the application displaying them.</p> |



The term `aeset` stands for anchored element set.

The rules for entity reference names are:

- ASCII characters a-z, A-Z, period (.), and hyphen (-) are allowed. The first character of a name is always in the set a- z or A-Z.
- A name can have a maximum of 12 characters.
- Entity names are not case-sensitive, which means that &body, &BODY, and &Body are equivalent.



Do not forget to precede the entity with an ampersand and close it with a semicolon, such as &F-TITLE; or &DEFINE:P;"BR".

The story entities are:

| Entity | Explanation |
|--------------------|--|
| &BODY; | Interpolate the story BODY section in place of the entity. [false if story body section is empty] [true if story body section is not empty] |
| &AASET; | Interpolate the story AASET section in place of the entity. [false if story aaset section is empty] [true if story aaset section is not empty] |
| &F-<name>; | Interpolate the named field from the story in place of the entity. If the entity is one of F-AIR-DATE, F-CREATE-DATE, F-MODIFY-DATE, F-MOS-DURATION, F-AUDIO-TIME, F-BACK-TIME, F-CUME-TIME, F-TOTAL-TIME, or F-TAPE-TIME, the field is output using the strftime function. Furthermore, a formatting string can follow these fields. The formatting string must immediately follow the ‘;’ of the entity and be enclosed in single or double quote characters. See “Optional Format Strings” on page 439 for more information. [true if the field is present in the story] [false if the field is not present in the story or is blank or empty] |
| &IF;<entity-name>; | If the entity yields a true condition, process template text up to a closing ELSE or ENDIF in the output stream. If the entity yields a false condition, skip over all template text up to the next ELSE or ENDIF (not output). Empty and blank fields are treated as a false IF condition. Only fields which are present and contain at least one non-blank character cause a true IF condition. |

| Entity | Explanation (Continued) |
|----------|---|
| &ELSE; | Terminates the first part of an IF condition. If the IF condition was “true,” it is now treated as “false” and all template text up to the closing ENDIF is skipped. If the IF condition was “false,” it is now treated as “true” and all template text up to the closing ENDIF is processed. |
| &ENDIF; | Terminates an IF condition or an IF/ELSE combination. |
| &FIELDS; | Only used for test conditions [false if story fields section is empty or blank] [true if story fields section is not empty] |

NSML to HTML Conversion

The iNEWS system uses the News Story Markup Language (NSML) as its native story storage format. When HTML format is selected, the txnet program converts the NSML into HTML as stories are exported.

Various NSML tags can be redefined on export and translated to another code, if desired. For instance, it would be possible to translate the NSML paragraph tag into an HTML break tag (
) instead of the default HTML paragraph tag (<P>).

By placing DEFINE statements at the top of the HTML export template before the <HTML> line, the default conversion of NSML tags into HTML tags can be modified. Format of a DEFINE statement is: &DEFINE:<tag>;“<string>” or &DEFINE;<tag>;’<string>’.



Single or double quotes can be used.

The string includes all characters up to the matching delimiting quote; this allows inclusion of the other quotation mark within the string.

Standard definitions, which can be redefined, are listed below:

| Opening NSML Tag | Output String | Closing NSML Tag | Output String |
|------------------|---------------|------------------|---------------|
| BODY | “” | /BODY | “\r\n” |
| AESET | “” | /AESET | “\r\n” |
| P | “\r\n<p>” | /P | “</p>” |
| I | “<i>” | /I | “</i>” |
| U | “<u>” | /U | “</u>” |

| Opening NSML Tag | Output String | Closing NSML Tag | Output String |
|------------------|------------------|------------------|------------------|
| B | "" | /B | "" |
| PI | "<i>" | /PI | "</i>" |
| CC | "<i>" | /CC | "</i>" |
| A | "\r\n<!--a-->" | | |
| AE | "\r\n<!--ae-->" | /AE | "\r\n<!--/ae-->" |
| AP | "\r\n<p>" | /AP | "</p>" |
| MC | "\r\n<!--mc-->" | /MC | "\r\n<!--/mc-->" |
| PB | "\r\n<!--pb-->" | | |
| TAB | "\r\n<!--tab-->" | | |
| WP | "\r\n<!--wp-->" | | |
| RTL | "<rtl>" | /RTL | "</rtl>" |
| BG | "<bg>" | /BG | "</bg>" |
| LINK | "<link>" | /LINK | "</link>" |
| SHOW | "<show>" | /SHOW | "</show>" |
| URL | "<url>" | /URL | "</url>" |
| HIDDEN | "<hidden>" | /HIDDEN | "</hidden>" |

Here are some examples:


| Example | Purpose |
|--|--|
| &DEFINE:P;"output this" | Redefines the open paragraph conversion. |
| &DEFINE:/P;"found a close paragraph tag" | Redefines the close paragraph conversion |



The delimiting quotation mark must immediately follow the end of the entity, namely the semicolon.

Use \r and \n in the <string> to include a carriage return or a line feed.

The DEFINE statement can also be used in the HTML skeleton form to modify the display format for date or time fields.

| Statement | Explanation |
|---|--|
| <pre>&DEFINE:<type>; "<format>" or &DEFINE:<type>; '<format>'</pre> | <p>Define default formats. The format includes all characters up to the matching delimiting quote (this allows inclusion of the other quotation mark within the format).</p> <p>The <type> can be DATE or TIME.</p> <p>The default format for DATE is: "%D:%T".</p> <p>The default format for TIME is: "%M:%S".</p> <p>DATE applies to <date> fields. TIME applies to <duration> fields.</p> <p>Format strings are defined by the strftime function.</p> <p>You can still include a format string to override the default when referencing a <date> or <duration> fields individually, as in: &f-create-date; "%c"</p> |
| <p> <i>A lowercase c is recommended; an uppercase C would denote the two digit century, such as &f-create-date; "%C".</i></p> <pre>&DEFINE:<entity name>; "<string>"</pre> | <p>Define standard entity representations for each of the four characters: left and right angle brackets (<>), ampersand (&), and quotation mark ("").</p> <pre>&DEFINE:lt; "..."</pre> <pre>&DEFINE:gt; "..."</pre> <pre>&DEFINE:amp; "..."</pre> <pre>&DEFINE:quot; "..."</pre> <p>The default definitions are (respectively): &lt;, &gt;, &amp;, and &quot;.</p> |

SHOW and HIDE statements can also be used in the HTML skeleton to modify output.

| Statement | Explanation |
|----------------------------------|---|
| <code>&SHOW:A;</code> | <p>Causes output of tags for locating and referencing anchors and production cues (anchored elements). This is the default, which is the opposite functionality prior to introduction of this entity.</p> <p>The string produced whenever an NSML <code><a></code> tag is encountered in the “template” is:</p> <pre data-bbox="642 520 1249 546"><A NAME="A<id>" HREF="#AE<id>">[<id>]</pre> <p>For an NSML <code><ae></code> tag, the string produced is:</p> <pre data-bbox="642 616 1249 642"><A NAME="AE<id>" HREF="#A<id>">[<id>]</pre> <p>The <code><id></code> is the production cue identifier, typically a small number.</p> <p>This results in strings such as “[1]” appearing as HTML links in the text at the anchor point and also preceding the production cue. The two links point at each other.</p> |
| <code>&HIDE:A;</code> | <p>Suppresses output of HTML anchor tags.</p> |
| <code>&HIDE:COMMENTS;</code> | <p>Empty fields and non-present fields in the HTML output are noted with comments similar to the following:</p> <pre data-bbox="642 980 1249 1006"><!-- "var-1" field is empty --></pre> <pre data-bbox="642 1032 1249 1058"><!-- "appl-1" field not present --></pre> <p>If the <code>&HIDE:COMMENTS;</code> entity is processed, comments normally generated for the conditions “field not present” and “field is empty” will not be output to the HTML document.</p> <p>This does not affect comments generated from error conditions.</p> |
| <code>&SHOW:COMMENTS;</code> | <p>Generation of these HTML comments can be turned back on using <code>&SHOW:COMMENTS;</code>.</p> |

Web Story Directives

Any story can contain Web Story directives. Directives are included as text. This allows system administrators to embed HTML codes within stories. Without these directives, characters and symbols such as the greater than sign (`>`) would normally be translated to `>`; and not appear as HTML coding when exported to the Web server.

| Directive | Explanation |
|-------------|--|
| #HTMLSTART# | All text in the story after this directive is processed as literal text. Four characters—<, >, &, and ”— are output without being converted into HTML entities. Furthermore, all NSML tags are stripped from the input stream so they are not converted. However, to make viewing the HTML source a little easier, each paragraph closing tag is converted and output as a Carriage-Return / Line-Feed pair. |
| #HTMLEND# | This will terminate the processing of story text as literal text. |

Optional Format Strings

Optional format strings apply to date and time-related fields that you use for the field ID story entity references. You can add several time or date elements together (after the field ID) to create an optional format string, which changes the presentation of the time or date information field.



The optional time format strings must be enclosed within quotes in the HTML export template.

The time and date elements to use in an optional format string to modify date and time field ID areas may vary depending on the version of Linux. Use `info strftime` at the console to get the format directives.

The difference between %U and %W is which day is considered the first day of the week. Week number 01 is the first week in January, with a Sunday for %U or a Monday for %W. Week number 00 contains those days before the first Sunday or Monday in January for %U and %W, respectively.

Only certain fields—those relating to time and date—can recognize the optional format string, made up of the time and date elements. Specify these fields in field IDs, using the types shown here:

| FIELD TYPE (* cell type) | Description | Default Format |
|-----------------------------|--|----------------|
| AIR-DATE | The date and time a story airs, using the show timing function | %D%T |
| CREATE-DATE | Seconds since January 1, 1970 00:00:00 GMT | %D%T |
| MODIFY-DATE | Seconds since January 1, 1970 00:00:00 GMT | %D%T |

| FIELD TYPE (* cell type) | Description | Default Format |
|-------------------------------------|---|-----------------------|
| MOS-DURATION | Runtime, in seconds, of a MOS item that is inside the Story Form | %M:%S |
| AUDIO-TIME | Audio read time of the story in seconds. Normally based on read rate and word count, but can be entered by the user | %M:%S |
| BACK-TIME | Hard in-time of the story, in seconds | %M:%S |
| CUME-TIME | Hard out-time of the story, in seconds | %M:%S |
| Duration Control (*) | Number of seconds | %M:%S |
| Calendar Control (*) | Seconds since January 1, 1970 00:00:00 GMT | %D%T |



Duration Control and Calendar Control are possible cell types for data entry, not to be confused with the standard iNEWS field types provided in the rest of the previous table.

See [“Form Field Types and Definitions” on page 208](#) for more information.

If you use field names in the field IDs without creating an optional format string, default formats for the field types, as listed in the previous table’s “Default Format” column, are used.

For instance, you decide to include a story creation date in the HTML template. Instead of using defaults provided for the CREATE-DATE field (%D %T - date as %m/%d/%y and time as %h:%m:%s), you want the creation date information to appear differently.

- So, you should choose the following date and time elements:
 - %a – Abbreviated weekday name
 - %B – Full month name
 - %d – Day of month (01 - 31)
 - %Y – Year as ccy
 - %r – Time as %I:%M:%S [AM|PM]
- Then, enter elements in the optional format string, enclosing the string within quotes.

The field_ID story entity reference looks like this:

```
&f-create-date;"%a %B %d %Y %r"
```

The field_ID is embedded in HTML format tags. If the HTML export template is applied, the entry looks like this:

```
<b>CREATE-DATE:</b>&f-create-date;"%a %B %d %Y %r"
```


Data from the field would appear on the page similar to this:

CREATE-DATE: Sat July 10 2004 5:30:45 PM



Without the spaces between each element inside the quotation marks—such as, “%a%B%d%y%r”—the date will appear as one continuous word. For instance:

CREATE-DATE: SatJuly1020045:30:45PM

Sample HTML Export Template

The following sample shows a news story in an iNEWS HTML export template used to define the appearance of the story, and the resulting HTML output.

Characteristics of the Story

The story has the following characteristics:

- TITLE field with an associated value: TEASE-BREAK
- MODIFY-DATE field with an associated value: 862347754
- PRESENTER field with an associated value: SB*
- PAGE-NUMBER field with an associated value: 75
- Two production cues (each formatted in its own paragraph):

VO ENG

Coming Up Next...

- Story text:

(BELL)

STILL AHEAD ON EYEWITNESS NEWS...

THE MAXWELL CLUB HONORS SOME OF OUR AREA'S BEST HIGH SCHOOL FOOTBALL PLAYERS. YOU'LL MEET THEM... COMING UP.

Contents of the Export Template

An HTML export template is applied to the story. The template's contents look like this:

```
<html>
<head>
<title>&f-title;</title>
<h1>
&f-title;
</h1>
</head>
<body>
<!--This is an example of an iNEWS HTML template.-->
<table border="4" width="100%">
  <tr>
    <td colspan=1 align="left" width="10%">
      <b>PAG:</b>&f-PAGE-NUMBER;
    </td>
    <td colspan=1 align="left" width="40%">
      <b>MOD-DATE:</b>&f-modify-date;
    </td>
    <td colspan=1 align="left" width="50%">
      <b>PRESENTER:</b>&f-PRESENTER;
    </td>
  </tr>
  <tr>
<h2><th colspan=2 align="left" align="left">Story:
</th></h2>
```

```
</tr>
<tr>
  <td colspan=2 align="left" width="100%"> <p>&BODY;
</td>
</tr>
<tr>
<h2><th colspan=2 align="left" >Anchored Element List:</th></h2>
</tr>
<tr>
  <td colspan=2 align="left" width="100%">
    &A&ESET;
  </td>
</tr>
</table>
</body>
</html>
```

The next section shows the HTML output as a result of this process.

Resulting HTML Output

```
<html>
<head>
<title>TEASE/BREAK</title>
<h1>
TEASE/BREAK
</h1>
</head>
<body>
<!--This is an example of an iNEWS HTML form.-->
<table border="4" width="100%">
  <tr>
    <td colspan=1 align="left" width="50%">
      <b>PAG:</b>75
    </td>
    <td colspan=1 align="left" width="40%">
      <b>MOD-DATE:</b>04/29/97 14:02:34
    </td>
    <td colspan=1 align="left" width="50%">
      <b>PRESENTER:</b>SB*
    </td>
  </tr>
  <tr>
<h2><th colspan=2 align="left" align="left">Story: </th></h2>
  </tr>
  <tr>
```

```

        <td colspan=2 align="left" width="100%">
(BELL)
STILL AHEAD ON EYEWITNESS NEWS...
THE MAXWELL CLUB HONORS SOME OF
OUR AREA'S BEST HIGH SCHOOL FOOTBALL
PLAYERS. YOU'LL MEET THEM... COMING
UP.
        </td>
</tr>
<tr>
<h2><th colspan=2 align="left" >Anchored Element List:</th></h2>
</tr>
<tr>
        <td colspan=2 align="left" width="100%">
                <p><I>VO ENG</p>
                <p>Coming Up Next...
        </td>
</tr>
</table>
</body>
</html>

```

Publishing iNEWS Stories to the Web

After the Web Publishing configuration is complete you can publish stories to the Web.

To publish to the Web:

1. Select a story from the news database.
2. Merge pieces of the story with an HTML export template to recreate the story as an HTML document.
3. Use the txnet facility to put the HTML document onto another computer, known as the Web Publishing server.
4. Revise the HTML-formatted story, if necessary, to complete the publishing process.

Web Access

The Web Access software, which is normally installed with your iNEWS newsroom computer system, is used to:

- Access the database
- Format the story the user selects into Hypertext Markup Language (HTML), using the default HTML export template
- Present the story to the user's Web browser

Formatting of information sent to the user's Web browser is configurable by the system administrator. How the directory listing, queue contents, and stories are converted to HTML for presentation are governed by HTML templates inside the database.

The Web Server

The Apache Web server is loaded on iNEWS Servers as part of the Linux system, and automatically launches when those servers are booted. You can check to see if the Apache Web server is currently running on your system by pointing your browser to:

```
http://<servername>
```

The <servername> is the iNEWS Server name you normally log in to from the iNEWS Workstation. If the Web server is running, you should see a test page with the Apache logo.

Logging in via Web Access

You must have at least one Web session configured on your system to log in over the Web. The format of a Web session resource in the `/site/config` file is:

```
websession <session number>
```

For instance: `websession 401`

Each websession resource must be included in a reslist list in each appropriate host section of the `/site/config` file.

To log in to the iNEWS database with a Web browser:

- ▶ Enter a URL in the following format: `http://<servername>/cgi-bin/NewsWeb`

This will open a page containing a drop-down list of iNEWS Servers accessible via the Web. When you select one of those servers and click submit, you will be taken to a login screen. If you know the servers configured for Web Access, you may include their names in the URL to go directly to the login dialog page:

```
http://<servername>/cgi-bin/NewsWeb.nrcs-a
```

At the login dialog screen, enter your user name and password and click login. When logging in over the Web for read-only database access, group security on the system is respected. Users who do not have access to queues and directories protected by a read-group will not see those areas when they log in over the Web.



The message of the day screen is not displayed when logging in via a Web browser.

Web Access Story Templates

When users view a story in their Web browser, the system presents the story in HTML format. The appearance of this story is configurable by the system administrator via a Web Access template, similar to the HTML export templates used for Web publishing.

When accessing iNEWS over the Web, the story form assigned to a story is noted and a corresponding Web Access form is searched for under queues specified by the `Q_WEBACC_FORMS` entry in the system's queues dictionary. The default location for the Web Access form (HTML export template) is `SYSTEM.WEBFORMS`.

The appropriate Web Access form is sought from:

```
SYSTEM.WEBFORMS.<first letter of story form>.<story form>
```

For instance, if the form assigned to the story was called `script`, then the Web Access form—that is, the top HTML export template—in the queue, `SYSTEM.WEBFORMS.S.SCRIPT` would be used to display the story. Stories assigned the

rundown story form would be displayed with the Web Access form in SYSTEM.WEBFORMS.R.RUNDOWN. Stories assigned the memo story form would look for the top HTML export template in the queue, SYSTEM.WEBFORMS.M.MEMO, for their display on the Web browser.

If a corresponding Web Access form was not created, the system uses the HTML export template in the default Web Access queue as specified by the W_WEBACC_FORM entry in the words dictionary. The default value is ACCESS-FORM. So, a story assigned a form that does not have a corresponding SYSTEM.WEBFORMS queue will be displayed with the system-wide default HTML export template in SYSTEM.WEBFORMS.A.ACCESS-FORM.



If a system-wide default Web Access form has not been created, the story will be displayed with a hard-coded default.

The following Web Access form locations will be checked, in order, for an HTML export template to display stories:

- SYSTEM.WEBFORMS.<first letter of story form>.<story form>
- SYSTEM.WEBFORMS.A.ACCESS-FORM
- Hard-coded internal default

You may create HTML export templates to define the appearance of stories viewed via a Web browser. The export template is stored as an iNEWS database story whose text is in HTML with optional embedded references to story entities. The iNEWS system contains an HTML export template to use as the default, but you can customize it or create new templates, using story entities. A full listing of all entities available to Web Access forms used for database browsing appears in [“Template Entities” on page 451](#).



The same HTML export templates set up on your system for exporting stories in HTML format can also be used when browsing the database over the Web. See [“The HTML Export Template” on page 432](#) for more information.

Default Story Template

The following is a default story template provided with the iNEWS software.

```
<html>
<head>
<title>&F-TITLE;</title>
</head>
<body>
  <table border=4 width='100%'>
    <tr>
      <td colspan=1 width='50%'><b>CREATED BY:</b>&F-CREATE-BY;
    </td>
      <td colspan=1 width='50%'><b>CREATED:</b>&F-CREATE-DATE;
    </td>
    </tr>
    <tr>
      <td colspan=1 width='50%'><b>PAGE:</b>&F-PAGE-NUMBER;
    </td>
      <td colspan=1 width='50%'><b>MODIFIED:</b>&F-MODIFY-DATE;
    </td>
    </tr>
    <tr><h2>
      <th align=left colspan=2>Story:
    </th></h2>
    </tr>
    <tr>
      <td colspan=2 width='100%'>&BODY;
```

15 Web Publishing and Access

```
        </td>
    </tr>
    <tr><h2>
        <th align=left colspan=2>Anchored Element List:
    </th></h2>
    </tr>
    <tr>
        <td colspan=2 width='100%'>&A&ESET
    </td>
    </tr>
</table>
</body>
</html>
```

Web Access Directory and Queue Templates

The iNEWS system allows the system administrator to customize the display of directory listings and listings of a queue's contents via system-wide directory and queue HTML export templates rather than predefined formats. However, it is not necessary to customize directory and queue templates. Defaults can be used, if desired. See [“Default Directory Template” on page 456](#) and [“Default Queue Template” on page 458](#) for more information.

The template used for directory pages is located in the queue defined by the Q_WEBACC_FORMS and W_WEBACC_FORM definitions (SYSTEM.WEBFORMS.A.ACCESS-FORM). To distinguish this directory template from the story template, the directory template is identified by having the word directory as the first word in the index field (also known as the sortfield). Only a single template is required for directory page construction.

The template used for queue pages is located in the queue defined by the Q_WEBACC_FORMS definition and the queue form name assigned to the queue being displayed. This is similar to how story forms are handled. To identify the template as a queue template, the word queue must appear as the first word in the index field (also known as the sortfield). This allows both the queue template and story template with the same name to reside in the same queue.

If a queue template is not found using the queue form name of the queue, the system looks in the queue defined by the Q_WEBACC_FORMS and W_WEBACC_FORM definitions, similar to story templates. The queue template is identified with the word queue as the first word in the index field.

Put the system-wide default Web Access form (HTML export template) at the top of the SYSTEM.WEBFORMS.A.ACCESS-FORM queue. Place stories titled directory and queue beneath the form used by default to display story contents.

Template Entities

The following table lists all of the available entities that can be placed in either a story, directory, or queue template. The Q- and D- entities are only valid for Web Access. They are not available for Web Publishing. If an entity is not valid for a certain type of template, it is noted in the entity description.

Each entity has IF condition definitions included following the text handling description. If an entity is not valid for a certain type of template, it will always result in a false condition when tested. Empty and all blank fields produce a false IF condition. Only fields present in the story and containing at least one non-blank character cause a true IF condition.





Do not forget to precede the entity with an ampersand and close it with a semicolon, such as &F-TITLE; or &DEFINE:P;"BR".

| Entity | Explanation |
|---------|--|
| &BODY; | Interpolate story BODY section in place of entity. [false if story body section is empty] [true if story body section is not empty] |
| &AASET; | Interpolate story AASET section in place of entity. [false if story aaset section is empty] [true if story aaset section is not empty] |

| Entity | Explanation (Continued) |
|--------------------|---|
| &F-<name>; | <p>Interpolate the named field from the story in place of the entity.</p> <p>If the entity is one of F-AIR-DATE, F-MOS-DURATION, F-CREATE-DATE, F-MODIFY-DATE, F-AUDIO-TIME, F-BACK-TIME, F-CUME-TIME, F-TOTAL-TIME, or F-TAPE-TIME, the field is output using the strftime function.</p> <p>Furthermore, a formatting string can follow these fields. It must immediately follow the ‘;’ of the entity and be enclosed in single or double quotation marks.</p> <p>Default format for the *-DATE fields is “%D %T”. Default format for the *-TIME fields is “%M:%S”.</p> <p>[true if the field is present in the story]</p> <p>[false if the field is not present in the story or is blank or empty]</p> |
| &IF:<entity name>; | <p>If the entity yields a true condition, process template text up to a closing ELSE or ENDIF in the output stream. If the entity yields a false condition, skip over all template text up to the next ELSE or ENDIF (not output).</p> <p>Empty and blank fields are treated as a false IF condition. Only fields which are present and contain at least one non-blank character cause a true IF condition.</p> |
| &ELSE; | <p>Terminates the first part of an IF condition. If the IF condition was “true,” it is now treated as “false” and all template text up to the closing ENDIF is skipped. If the IF condition was “false,” it is now treated as “true” and all template text up to the closing ENDIF is processed.</p> |
| &ENDIF; | <p>Terminates an IF condition or an IF/ELSE combination</p> |
| &FIELDS; | <p>Only used for test conditions</p> <p>[false if story fields section is empty or blank]</p> <p>[true if story fields section is not empty]</p> |
| &REPEAT; | <p>Bracket a section of the template text, up to an ENDREPEAT entity. The bracketed text will be processed for each object to be displayed on the page. This is only valid for directory and queue templates. If there are no objects to be displayed, the directory or queue is empty, and all of the bracketed text is skipped over.</p> |
| &ENDREPEAT; | <p>Terminates a REPEAT section. If there is another subdirectory or queue entry to be displayed on the page, template processing will continue at the preceding REPEAT entity. If there are no more objects to be displayed, template processing continues at the point following this entity.</p> |

| Entity | Explanation (Continued) |
|---------------|--|
| &D-CHILD; | Interpolate current subdirectory name in place of the entity. Not valid for queue or story templates. [always true] |
| &D-COUNT; | Interpolate number of sub-directories contained in current directory. This is intended primarily for test conditions, to determine if a directory is empty. Not valid for queue or story templates. [false for empty directory] [true for directory with at least one entry] |
| &D-CURRENT; | Interpolate a link reference, such as HREF, to current sub-directory entry in place of the entity. Not valid for queue or story templates. [false for an empty directory] [true for a directory with at least one subdirectory] |
| &D-DIRECTORY; | Nothing, only used for test condition. Not valid for queue or story templates. This is the opposite of D-QUEUE. [true if the current subdirectory is a directory] [false if the current subdirectory is a queue] |
| &D-LOGOUT; | Interpolate a link reference, such as HREF, which includes a Web Access server logout request in place of the entry. [always true] |
| &D-NAME; | Interpolate the current directory name in place of the entity. [always true] |
| &D-PARENT; | Interpolate a link reference, such as HREF, to the parent directory of the current directory in place of the entity. [false when current directory is the root directory] [true when current directory is not the root directory] |
| &D-QUEUE; | Nothing, only used for test condition. Not valid for queue or story templates. This is the opposite of D-DIRECTORY. [true if the current subdirectory is a queue] [false if the current subdirectory is a directory] |
| &D-SERVER; | Interpolate the server name (computer name) in place of the entity. [always true] |

| Entity | Explanation (Continued) |
|---|---|
| &Q-COUNT; | <p>Interpolate number of queue entries contained on current page. This is intended primarily for test conditions, to determine if a queue is empty. Not valid for directory templates.</p> <p>[false for empty queue]</p> <p>[true for queue with at least one entry]</p> <p>[always true for story template]</p> |
| &Q-CURRENT; | <p>Interpolate a link reference, such as HREF, to current queue entry in place of the entity. This includes the complete URL-string to identify the current story position within the queue. Not valid for directory templates.</p> <p>[false for empty queue]</p> <p>[true for queue with at least one entry]</p> <p>[always true for story template]</p> |
| &Q-HREF; | alias for Q-CURRENT (preferred name) |
| &Q-LOGOUT; | <p>Interpolate a link reference, such as HREF, which includes a Web Access server logout request in place of the entry.</p> <p>[always true]</p> |
| &Q-MODIFY-DATE; | <p>Interpolate modified time of current queue entry in place of the entity. Not valid for directory templates. The modified time is formatted using the strftime function with a default format definition of “%D %T”. This format can be redefined via the DEFINE:DATE entity described below. Just as with *-DATE fields, you can supply a format following this entity.</p> <p>[always true]</p> |
|  <i>This is the queue entry modified time and NOT the story modified time.</i> | |
| &Q-NAME; | <p>Interpolate current queue name in place of the entity.</p> <p>[always true]</p> |
| &Q-NEXT; | <p>Interpolate a link reference, such as HREF, to the next queue entry within the queue. This includes the complete URL-string to identify the next queue entry position within the queue. Not valid for directory templates.</p> <p>[false in queue template when current page contains last queue entry]</p> <p>[true in queue template when current page does not contain last queue entry]</p> <p>[false in story template when current story is the last story in the queue]</p> <p>[true in story template when current story is not the last story in the queue]</p> |

| Entity | Explanation (Continued) |
|---|--|
| &Q-PARENT; | Interpolate a link reference, such as HREF, to parent directory of current queue in place of the entity. [always true] |
| &Q-PREV; | Interpolate a link reference, such as HREF, to the previous queue entry within the queue. This includes the complete URL-string to identify the previous queue entry position within the queue. Not valid for directory templates. [false in queue template when current page contains first queue entry] [true in queue template when current page does not contain first queue entry] [false in story template when current story is the first story in the queue] [true in story template when current story is not the first story in the queue] |
| &Q-QUICK-INDEX; | Interpolate quick index contents of current queue entry. Not valid for directory templates. [always true] |
|  | <i>Twenty (20) characters are always interpolated. This is different than story field contents where only the text present is interpolated.</i> |
| &Q-SERVER; | Interpolate the server name (computer name) in place of the entity. [always true] |
| &Q-STORY-TIME; | Interpolate story read time of current queue entry in place of the entity. Not valid for directory templates. The modified time is formatted using the strftime function with a default format definition of “%M:%S”. This format can be redefined via the DEFINE:TIME entity described below. Just as with *-TIME fields, you can supply a format following this entity. [always true] |

See [“NSML to HTML Conversion” on page 435](#) for more information.


```
&IF:D-COUNT;  
&REPEAT;  
  <a href=&D-CURRENT;>  
    &IF:D-QUEUE;  
      <img src='/icons/text.gif' hspace=5 border=0>  
    &ELSE;  
      <img src='/icons/folder.gif' hspace=5 border=0>  
    &ENDIF;  
    &D-CHILD;  
  </a>  
  <br>  
&ENDREPEAT;  
&ELSE;  
  <img src='/icons/blank.gif' hspace=5>[Empty]<br>  
&ENDIF;  
</body>  
</html>
```


Web Access Configuration

Error messages returned by the Web Access server can be adjusted by editing a system dictionary. In addition, various default parameters can be adjusted.

Configuration parameters for the Web Access server must be included in the file `/site/web/configuration`, accessible by the Web Access server on the computer on which the Web Access server is running. This file is read once when the Web Access server starts.

| Parameters | Definition |
|--|---|
| <code>announce attempted logins=<value></code> | If the value is non-zero, all unsuccessful logins will be printed on the console. This includes failures due to an unknown user name, an invalid password, or a blacklisted user. The default is zero (0). |
| <code>announce logins=<value></code> | If the value is non-zero, all logins will be printed on the console. The default is zero (0). |
| <code>diagnostics=<value></code> | This is primarily for debugging purposes. The value is a bitmask, which controls printing of diagnostics. For instance, if the value is odd, such as the 1 bit is on, a diagnostic is printed every time a session is requested but all session licenses are in use. The default is zero (0). |
| <code>hard expire = <seconds></code> | Number of seconds of inactivity which will force a reauthentication by the user and release the Web session for another user to log in to. The default is 1200 (20 minutes). |
| <code>putenv=<environment string></code> | Allows environment settings to be put into the initial environment. The environment string should be of the form <code>name=value</code> . This will be primarily used for debugging purposes. |
| <code>queue entries per page = <number></code> | Number of stories to put on a queue page. This can be a number from 1 to 50. The default is 50. |

| Parameters | Definition (Continued) |
|--|--|
| <code>soft expire = <seconds></code> | Number of seconds of inactivity before the session license can be reused for another session. This will only affect a user if the server gets a session request and does not have any free session licenses. The server attempts to obtain the oldest session license—that is, the one that has been inactive the longest. If that session has been inactive for more than the soft expire interval, it will be reused. (Anytime a session is inactive for more than the hard expire interval, the user will be required to reauthenticate the session.) The default is 300 (5 minutes). |

The following definitions must be present in the file `/site/web/strings`, accessible by the Web Access server on the computer on which the Web Access server program is running. The format of this file is the same as all dictionary files. This file is read once when the Web Access server starts. Definitions can be localized, as necessary.

| | |
|--------------------------------------|---|
| <code>session error</code> | <code>/iNEWS news session error</code> |
| <code>session login</code> | <code>/iNEWS news session login</code> |
| <code>session terminated</code> | <code>/Your iNEWS news session has been successfully terminated!</code> |
| <code>offline</code> | <code>/iNEWS news is offline.</code> |
| <code>try again later</code> | <code>/Please try again later.</code> |
| <code>cannot set subdirectory</code> | <code>/Cannot set subdirectory.</code> |
| <code>cannot open dir file</code> | <code>/Cannot open directory file.</code> |
| <code>cannot open queue</code> | <code>/Cannot open queue.</code> |
| <code>cannot access</code> | <code>/Cannot access</code> |
| <code>invalid certificate</code> | <code>/Invalid session certificate</code> |
| <code>no license</code> | <code>/No Web Access licenses available.</code> |
| <code>name</code> | <code>/Name:</code> |
| <code>password</code> | <code>/Password:</code> |
| <code>login</code> | <code>/Login</code> |
| <code>clear login</code> | <code>/Clear Form</code> |

| | |
|---------------------------|---|
| enter login | /Please enter your name and password |
| reauthenticate | /iNEWS news Web session reauthentication |
| session expired | /Your iNEWS news session authentication has expired |
| reenter login | /Please re-enter your user name and password. |
| not authorized | /You are not authorized to access the iNEWS database |
| bad login | /The user name and/or password you entered are not valid. |
| no login see admin | /If you do not have a login, see your system administrator. |
| story unavailable | /You cannot access this story. |
| cgi short post | /input length less than CONTENT_LENGTH |
| cgi empty post | /CONTENT_LENGTH is missing |
| cgi empty query | /QUERY_STRING is null |
| cgi unknown request | /unknown REQUEST_METHOD |
| cgi expression error | /Expression error |
| cgi upload error | /Create of upload file failed |
| cgi no request | /REQUEST_METHOD is missing |
| ; | |
| ;Used by "NewsWeb" script | |
| ; | |
| no servers | /There are no iNEWS Servers configured. |
| server session | /iNEWS Server selection |
| server error | /iNEWS Server selection Error |
| server admin | /See your system administrator |
| server submit | /submit |
| server select | /Please select an iNEWS Server and click on "submit." |

16 iNEWS Projects

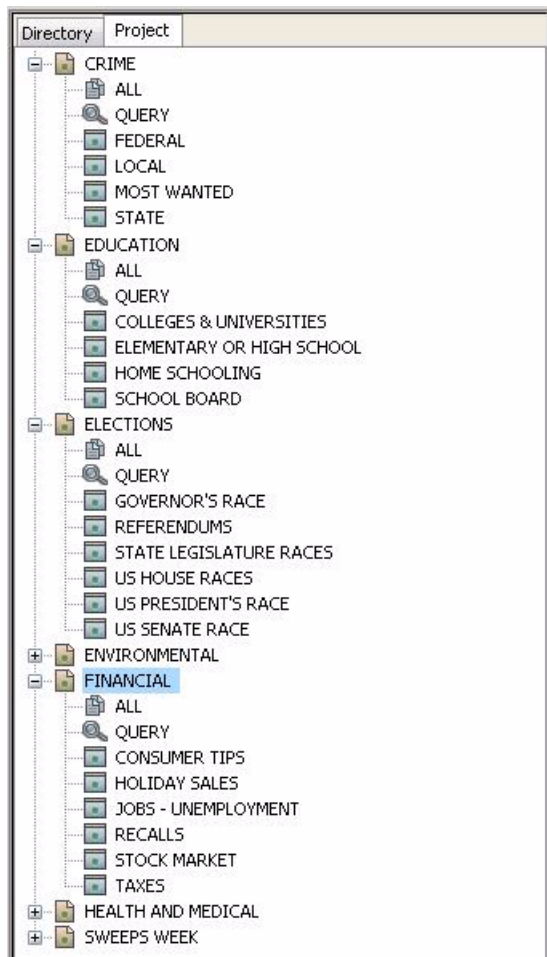
The iNEWS Project feature is a way of categorizing stories by topic so that news teams working on a particular topic can find everything related to it in a single place, without moving or copying the original source information from its current location in the database. Facets are sub-topics, providing additional granularity to projects. This chapter covers the setup and use of projects and facets within an iNEWS system.

This chapter contains the following main sections:

- [Overview of Projects and Facets](#)
- [Setting up the iNEWS Database for Projects](#)
- [Creating Projects](#)
- [Creating Facets](#)
- [Associating Stories with Projects or Facets](#)

Overview of Projects and Facets

At an iNEWS Workstation, projects appear in a tree-style structure in the Directory panel via the Project tab. Like directories, projects are expandable, which lets users view each project's ALL queue, QUERY queue, and sub-topics, known as facets. The following graphic shows a variety of sample projects and facets.



For instance, the Financial project has facets for topics like Recalls, Stock Market, and Taxes.

Every project has an ALL queue that displays in the Queue panel all stories associated with the project and its facets. Any indexed story can be associated with a project or facet. Stories associated with a project retain their original source permissions; therefore, a user without read access to a story's source queue will not be able to see that story in a project to which its associated even if the user has read access to the project.

Every project has a QUERY queue—identified by the magnifying glass icon—which is the search queue that runs the project's query. For more information about search queues, see XREF.



*System administrators can view a list of existing projects and facets in the system from the console by typing, **list p**. Projects and facets are distinguished from each other in the list by the letters **P** (for project) and **F** (for facet). A question mark (?) is used to indicate projects or facets belonging to other systems, which cannot be used by the current iNEWS system.*



Do not use the `dbdump` command to transfer projects across systems. The `dbdump` command should only be used for project backup.

Setting up the iNEWS Database for Projects

Before users can benefit from the Projects feature, the iNEWS database must be set up to properly handle projects and their facets.

To set up iNEWS for projects:

1. Run `dbgen p` to create the project file.
2. Run `dbgen m` to re-initialize the mailbox to handle additional updates that come with projects.
3. Modify user accounts to enable the creation of new projects and facets by certain users. Such permissions may be assigned from the console or an iNEWS Workstation:
 - a. From the console, use the `utraits` command in the following format:


```
utraits <user> +mp
```

 For instance, type: `utraits jdoe +mp`
 - b. At an iNEWS Workstation, select **Tools > Options > Users** and search for the user(s) you want to permit to modify projects. Then click the **Modify** button and from the **Modify User Account** dialog box, select the **Manage Projects** check box located in the **Session Features** section of the dialog box.

The screenshot shows the 'Modify User Account' dialog box with the following details:

- User ID:** jdoe
- User Name:** Jane Doe
- Type:** Superuser, Blacklisted, Simplified, Local Only
- Edit Mode:** Insert, Overwrite
- Queues:** Home: PEOPLE.J.JDOE.NOTES, Destination: PEOPLE.J.JDOE.NOTES, Mail: PEOPLE.J.JDOE.MAIL
- Read Rate:** 0
- Session Features:** Media Browse, Connect Services, Manage Projects
- Configuration Features:** Toolbars, Color Highlights, Highlight Read Stories
- Queue Features:** Reorder Stories, Create/Kill Folders/Queues, Kill All Stories
- Password:** Force Change
- Buttons:** User Preferences..., OK, Get from Template, Cancel

Creating Projects

Before you can create projects, you must be logged in to an iNEWS Workstation as a user with proper permission (user trait) to manage projects.

To create a project:

1. Select the Project tab.
2. Select Project > New Project from the menu bar.

The Create New Project/Facet dialog box appears.

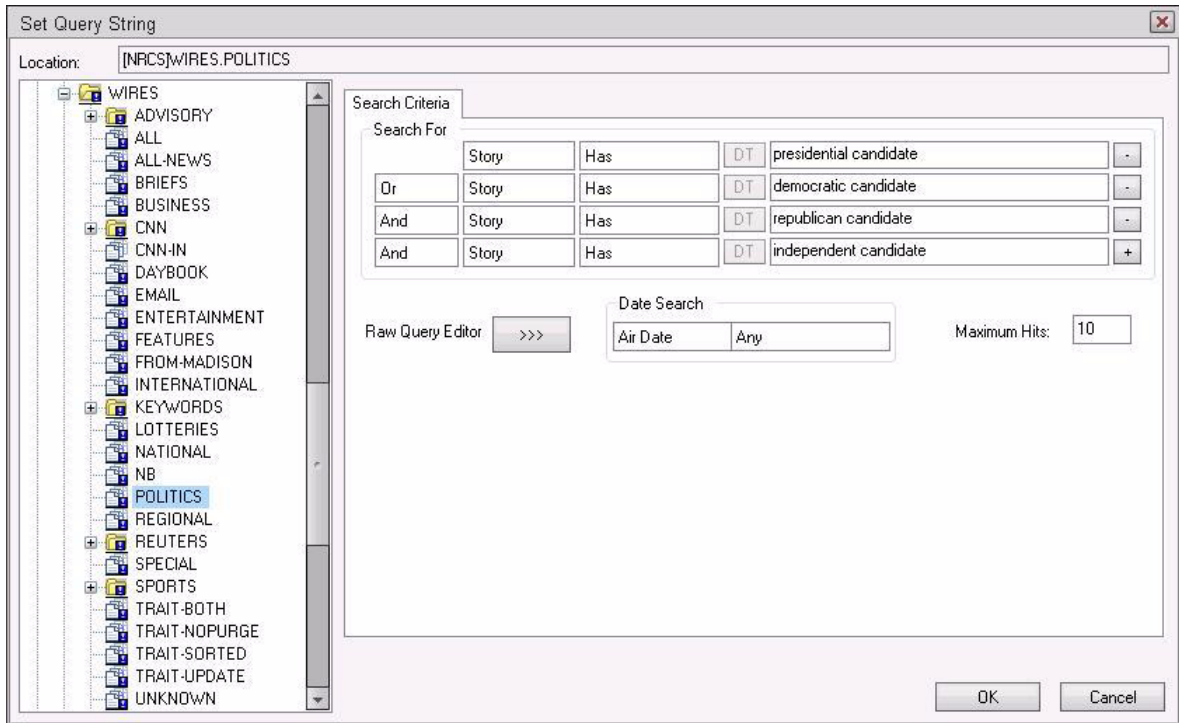
3. Type the name of the project. Project names can be up to 40 characters in length and can contain spaces, hyphens (-), or ampersands (&), but they cannot contain periods.



You cannot create a facet for the project until after the project itself is created. Therefore, the Facet and Facet Form fields will appear disabled when creating a project.

4. (Optional) Enter a description of the project.
5. Click Modify to create a query for the project.

The Set Query String dialog box appears in which you can select the indexed queue or queues you want to query and the search criteria you want to use.



For instance, if you are creating a project for your planned news coverage of upcoming elections, you can create a query to search through political wires for various candidates.

To select multiple queues, hold the Control key down and click two or more indexed queues you want to query.

You can use the Search For area of the Search Criteria tab to define the query, or write it out using the Raw Query Editor. The Date Search area lets you define a time frame for the query; you can search by the date a story aired or was created, or by last date a story was modified. Maximum hits can also be set in the Set Query String dialog box.

6. Click OK. Your search criteria is saved and the Set Query String dialog box closes.
7. In the Create New Project/Facet dialog box, continue the configuration by setting a read group, if any. The default is ! <none>. If set to ! <none>, no read group is defined; therefore, no restriction is applied to the project, so all users will be able to see the project and its facets.
8. Set the start and end date, if necessary. These dates define the date range in which the project appears in iNEWS Workstations. Projects will no longer appear after midnight (00:00) on the morning of the end date. Projects cannot have an end date prior to the start date.



Users can view expired projects and facets by right-clicking on one and selecting Show Expired Projects.

9. Define which forms should be used for the ALL and QUERY queues. For more information about queue forms, see [“Forms” on page 197](#).
10. The Notify Users box defines who receives notifications when project stories are added or modified. These notifications appear in the status bar of iNEWS Workstations. As the project administrator creating the project, you can select individual users or groups of users to add to the Notify Users list, but after the project is created, users with read access to the project can subscribe to receive notifications by right-clicking on the project in the Directory panel.
11. After you’ve completed setting up your project, click OK to save it. The project will appear on the Project tab in the Directory panel.

Creating Facets

Facets are sub-topics within a project that enable additional granularity in organization. You can create a facet after the parent project to which that facet will belong is created.

To create a facet:

1. Select the Project tab.
2. Right-click on the parent project.
3. Select New Facet.

The Create New Project/Facet dialog box appears with some information already filled in and displayed as read only.

For instance, the following example shows the dialog box as it would appear when creating a new facet for a project called Elections.

4. Type the name of the facet in the Facet field. Facet names can be up to 40 characters in length and can contain spaces, hyphens (-), or ampersands (&), but they cannot contain periods.
5. Set the start and end date, if necessary. These dates define the date range in which the facet appears in iNEWS Workstations. Facets will no longer appear after midnight (00:00) on the morning of the end date. Facets cannot have an end date prior to the start date. Facets also cannot have start or end dates beyond those defined by their parent projects.



Users can view expired projects and facets by right-clicking on one and selecting Show Expired Projects.

6. Define which queue form should be used for facets. For more information about queue forms, see “Forms” on page 197.

7. The Notify Users box defines who receives notifications when facet stories are added or modified. These notifications appear in the status bar at iNEWS Workstations. As the project administrator creating the facet, you can select individual users or groups of users to add to the Notify Users list, but after the facet is created, users with read access to the project can subscribe to receive notifications by right-clicking on the facet in the Directory panel.
8. After you've completed setting up your facet, click OK to save it. The facet will appear on the Project tab in the Directory panel.

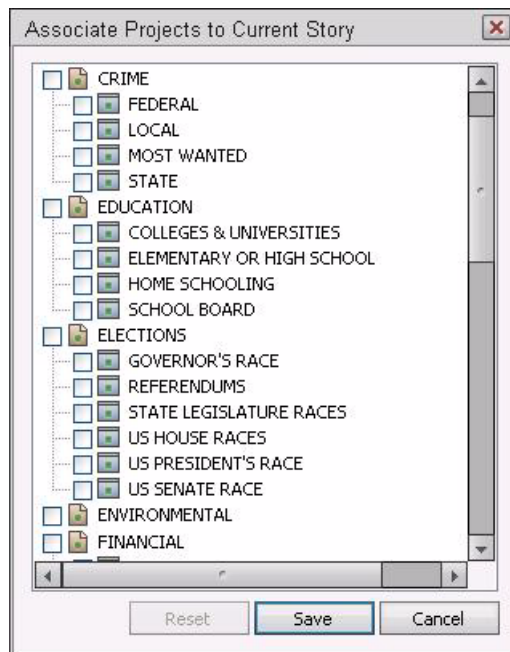
Associating Stories with Projects or Facets

Any indexed story can be associated with a project or facet.

To associate an indexed story to a project or facet:

1. From an iNEWS Workstation, navigate to an indexed queue in the Directory panel, and open it.
2. In the Queue panel, right-click on the story and select Associate Project/Facet.
3. When iNEWS requests confirmation that you want to edit the story, click Yes.

The Associate Projects to Current Story dialog box appears.



4. Select the project(s) or facet(s) to which you want to associate the story.
5. Click Save.

If a story is associated with a project or facet, a Project tab is displayed in the Story panel next to the Story tab when a user opens that story. The Story tab appears on top by default and displays the body of the story as usual, including production cues and machine instructions, if any. Users can click a story's Project tab to view the projects and facets with which that story is associated, along with related stories, if any.

| Associated Projects | | Related Stories | |
|--------------------------|--|-----------------|----------|
| Project/Facet | | Title | Location |
| ELECTIONS | | | |
| ELECTIONS.US HOUSE RACES | | | |



Users can also use the Shift+ALT+Left or Right Arrow keystroke combinations to navigate between the Story and Project tabs.

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 [476](#).

This appendix contains the following main sections:

- [Programs Invoked by iNEWS](#)
- [Commands Used by Avid Personnel Only](#)
- [Linux Commands Used in iNEWS](#)
- [Console Control Commands](#)
- [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 | detach | number |
| bdump | diddle | qcheck |
| binhex | edit | qstampcheck |
| biodebug | finit | qxcheck |
| biosleep | ftsdebug | rcat |
| biostat | ifis | strerror |
| catcheck | ifmaster | sysname |
| ccuputkey | ifsu | traverse |
| cmrdebug | keycheck | userclean |

| | | |
|---------|-----------|------------|
| dbcopv | kwdcheck | wordbreaks |
| dbgen | link | workdebug |
| dblinks | msgdebug | wxlate |
| dblock | netterm | xi |
| dbsize | nsupgrade | |

Linux Commands Used in iNEWS

The following Linux commands are available in iNEWS. For further 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 Control Commands

The table below lists available console control commands and their functions.

| Keyword | Explanation |
|---|---|
| bottom | Moves you to the newest (bottom) line in the history. |
| computer [<i>computer-name(s) *</i>] | Selects one or more of your system's servers, so that you can type a command on the selected server(s). The command may be abbreviated: typing <code>c</code> or <code>computer</code> plus the computer name(s) of the servers you want to select will work. Type the command plus an asterisk (*) to select all servers, such as <code>c *</code> . |
| down [<i>number-of-lines keyword</i>] | <p>Moves you that many lines forward (down) in the history.</p> <ul style="list-style-type: none">• <code>down number-of-lines</code> moves you that many lines forward in the history. For example, typing <code>down 30</code> moves you forward 30 lines. If you follow down with a number greater than the number of lines between your current position and the last line in the history, the down command moves you to the last (newest) line in the history.• <code>down keyword</code> searches forward through the history from your current position for that word. For example, to search forward for a line containing the word, <code>list</code>, type: <code>down list</code>. <p>The down command is not case sensitive. If you specify <code>PEOPLE</code>, <code>people</code> is a valid match. If the system does not find the keyword before reaching the bottom of the history, the console beeps and you are returned to the current line.</p> <ul style="list-style-type: none">• <code>down</code> with no parameter moves you down one line. <p>A wildcard character (#) can be used to match any character or to a search for a number. For instance, <code>down 160#</code> will search for the number 160 instead of moving down 160 lines.</p> |
| Exit Function | <p>To leave the console program and return to the MS-DOS prompt:</p> <ol style="list-style-type: none">1. Press the Command key.2. Press Ctrl+E.3. Press Enter. <p>(You cannot type <code>exit</code> at the Command prompt.)</p> <p>Type <code>console</code> at the prompt to restart the console program.</p> |

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

| Keyword | Explanation |
|--|---|
| <code>print</code> [<i>number-of-lines</i> <code>all</code>] | <p>Displays a number of lines of the history on the console screen.</p> <ul style="list-style-type: none"> • <code>print</code> followed by a number displays that many lines of the history beginning at the current line. • <code>print all</code> displays everything from the current line to the newest line. • <code>print</code> with no parameter displays the current line. <p>To cancel a <code>print</code> command while it is displaying console history, press any key. The console stops at the line last displayed on the screen, and your position in the history moves to that line.</p> |
| <code>reset</code> | <p>Attempts to unfreeze one or more of the console's I/O ports, if communication has failed between the console and your system's servers.</p> <p>To use <code>reset</code>, select the servers that are affected, press the Command key, and type the command. The command may be abbreviated: typing <code>r</code> or <code>reset</code> will work.</p> |
| <code>top</code> | <p>Moves you to the oldest line in recent console history. To see history older than this, use the <code>view</code> command to view a history disk log file.</p> |
| <code>up</code> [<i>number-of-lines</i> <i>keyword</i>] | <p>Moves you backward (up) some number of lines in the history.</p> <ul style="list-style-type: none"> • <code>up</code> <i>number-of-lines</i> moves you that many lines back in the history. For example, typing <code>up 30</code> moves you back 30 lines. <p>If you follow <code>up</code> with a number greater than the number of lines between your current position and the first line in the history, the <code>up</code> command moves you to the first line in the history.</p> <ul style="list-style-type: none"> • <code>up</code> <i>keyword</i> searches backward through the history from your current position for that word. For example, to search backward for a line containing <code>dbpurge</code>, type <code>up dbpurge</code>. <p>The <code>up</code> command is not case sensitive. If you specify <code>PEOPLE</code>, <code>people</code> is a valid match. If the system does not find the keyword before reaching the top of the history, the console beeps and you are returned to the current line.</p> <ul style="list-style-type: none"> • <code>up</code> with no parameter moves you back one line. <p>A wildcard character (<code>#</code>) can be used to match any character or to a search for a number. For instance, <code>up 160#</code> will search backwards for the number 160 instead of moving up 160 lines.</p> |

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



For commands that include an individual server letter (A, B, C, or D), use the pound (#) character to send to multiple servers without typing them individually; The console will convert the # to the server's letter, such as A or B. For instance, if you type connect #, the console sends the command connect A to server A, connect B to server B, and so forth. To enter a # without it converting to a server letter, press the Enter key on the numeric keypad, then press Shift+3.

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.*

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, 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> |
| maxhits=<number> | wordlength=<number> |
| min_passwd_length=<number> | |

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.

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 .
```

dbclose

```
dbclose
```

Closes the database.



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

dbdev

dbdev

Reports the disk partitions in use and the number of blocks allocated for the database on each disk partition. To find out the size of your database, type: **dbdev**

A message similar to the following appears

```
/dev/rp5  0          60000
/dev/rp6  60001     120123
/dev/rp7  120124    999999
```

The numbers reported are the first and last blocks for each partition. The last number of the report (999,999 in the example above) is the size of the database.

dbdump

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

Dumps individual stories or the entire database to tape.

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

| 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) |
| 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 |

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

To dump a queue to a new tape, add -n and the queue name to the command. For instance, to dump the queue SCRIPTS.JUNE.01 to a new tape, type: **dbdump c -n scripts.june.01**

By replacing the c with an a, you can add a queue to a tape without erasing information already on the tape. For instance, to append SCRIPTS.JUNE.10 to a tape, type:

```
dbdump a -n scripts.june.10
```

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

A message similar to the following appears:

```
data base size 4106240, free 72600=18%, freelist 2904
```



The freelist is compressed, so there can be as many as 5000 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 | f | n | q | s | v | +S | -S | -O] <pathname>
```

Checks the database for story errors.

| Valid Keys | Description |
|------------|--|
| b | Story block count only (no checking) |
| c | Complete check |
| 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 |
| q | List queue names |
| s | Skip queues that are skipped by <code>dbdump</code> |
| +S | Fix script related errors |
| -S | Skip (not report) script related errors |
| v | Verbose output |
| w | Check word counts |
| x | For <code>qstamp</code> 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 dblink 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 dbriginal command to reclaim space when the system is low on space.

The dbriginal 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 dbriginal 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 dbriginal 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 <path> parameter. Purges all queues with the default interval |
| -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 |
| 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 |
| -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>] |
| -N | Read from network socket for dump data |

dbrestore Command Key Options

| | |
|---------------|---|
| -p <queue> | Only with key letter s, will create queue |
| -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.

For instance, on a maximum 16 gigabyte database, type: **dbserver 16000000**

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>
```

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

| | |
|----------------------------|--------------------|
| changeform or cform | queueform or qform |
| displaylines | readgroup or rg |
| editorialgroup or eg | reindex |
| ftsindex | save |
| interplay or ip | sortfield or sf |
| mailbox or mb | storyform or sform |
| notify or ng | stripform |
| purgeinterval <days.hours> | writegroup or wg |

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`*

dbtraits Command Modes

| | |
|---------|--------------------------|
| f | Forms allowed |
| g | General |
| i | Inverted |
| index | Indexed |
| o | unorder (- only) |
| p | Printable |
| q | Queue operations allowed |
| r | Read access |
| refresh | Queue refresh |
| s | Sequential |
| so | Sorted |
| t | Text timing clocks |
| u | Update |
| w | Watch appends |
| x | Skip |

dbvisit

```
dbvisit -<d | v | i> -[r | m name] [-s] [block# ...]
```

dbvisit Command Options

| | |
|----|---|
| -d | Display progress by printing dots |
| -i | Validate ISAM file record and key sizes |
| -m | Machine name to disconnect (for online use) |
| -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 is a validation of ISAM key and record sizes to prevent problems after software upgrades when the key and/or record sizes of any ISAM files changed for the new software release.

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

There are two ways to run dbvisit:

- Without the -m option: The system must be offline and shut down. No one can log in until dbvisit is complete.
- With the -m option: Specify a machine that will be running dbvisit. The system must be offline and shutdown only long enough to enter the dbvisit command. Remaining machines can be brought up for users to log into. After the dbvisit procedure is complete, the machine running dbvisit can be reconnected following the normal procedure.



Before you run dbvisit, ensure the system is offline and the system is shut down. 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 -d
-----removed for documentation brevity-----

14:57:22 traversing directory

14:57:22 (each dot = 1 queue; each asterisk = 1 search queue; each colon
= 1,000 queue entries)

14:57:22 **.....:10:::::20:::::30:::::40::
14:57:50 :::::50:::::60:::::70:::::80:::::90::
-----removed for documentation brevity-----
```

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]
```

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 |
| + | Writes to disk (cannot be repaired by running <code>diskclear u</code>) |
| -- | Read database blocks |
| -<number> | The number of blocks to process at one time |
| u | Unclear database |

diskcopy

```
diskcopy -[<# of simultaneous copies>]
```

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

Copies database from master computer (usually server A) to a replacement computer. Enter it on the replacement server.

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

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] <queue> [file...]`

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 655](#). 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] <pathname>
```

Add the specified directory or queue 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.

help

```
help <command name>
```

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

A message similar to the following appears:

usage:

```
list configuration    [<termid> | <name>...]
list directory       [<name>...]
list terminal        [<termid> | <name>...]
list user            [<name>...]
list queue <name>    [<record limit>]
```

hogs

```
hogs [<pathname>]
```

Scans the directories or queues you specify and displays usage information for them. 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 [options] queue [- (a b d f g m s v)] | [<name> [<record limit>]] |
| list session [- (p v V)] | [<termid> <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)XWFi 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 | rnwg)=<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

Queue Options

qindex=<index value>
 -a address
 -b backwards
 -d deleted
 -f form
 -g groups
 -m modified by
 -s stamp
 -v verbose

Session Modifiers

-p process id

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

User Options

blacklist=(b | -)

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

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-[a | f | g | o | s | u | v] [<directory name>...]

list d Command Options

| | |
|-----|------------------------------------|
| d-a | Abstract printing traits |
| d-f | Queue and story form names |
| 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

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

# list d-f dead

SRPlo-LIsUGQsXWFiT queue form story form directory
Q-R-----I--G--X----                                DEAD

# list d-v dead

SRPlo-LIsUGQsXWFiT 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-LIsUGQsXWFiT 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] <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 |

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-WObfpRmF f.id time modified-time
25 h-disd ---M----- 13735 1 May2 17:07 2000
```

The index value consists of the selected sort field of the story you want to list. The quick index (qindex) value is optional, but must be a single word, and is not case-sensitive.

For instance, to get information for a story called “Nomad” in the queue PEOPLE.SMITH.NOTES, type:

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

list s

```
list s -[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.

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  SOEKCvTH sc  queues
danielmi  180  0   n   i  -OEKCvTH sc  dest: PEOPLE.D.DANIELMI.NOTES
                                     home: PEOPLE.D.DANIELMI
                                     mail: PEOPLE.D.DANIELMI.MAIL
```

```
NRCS-A$
```

The flags (`rr kb su m SOEKCvH 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 | s | Can Configure Shortcut Toolbar |
| O | Can Order | c | Can Configure Colors |
| E | Can Enter & Remove | | |

The letters in the header are defined as follows:

| | |
|----------|---------------------------------|
| u-h | Home queue |
| u-l | Last login |
| u-m | Mail queue |
| u-p | Preferences |
| u-r | Real name |
| u-t | Tracking information |
| u-v | Verbose output |
| <option> | list blacklist=[b -] user |
| | list "created>date1<date2" user |
| | 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 ...
```

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

| | |
|--|---|
| <code>v</code> | 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. |
| <code>i</code> | Install. To install the shift table into files you specify in the filename list. |
| <code>a</code> | Ask. To confirm installation of each file as you are installing the shift tables. Forces installation. |
| <code>p</code> | 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. |
| <code>f</code> | File. Specify with a <code><shift-file></code> filename, to use a file other than the default file for the case-shifting dictionary. The default file is <code>(/site/dict/shift)</code> . |
| <code>r</code> | Recursive. (For directories.) |
| <code><shift-file></code> | When you specify the <code>-f</code> option, enter the case-shifting dictionary name you want to use instead of the <code>/site/dict/shift</code> file. |
| <code><file1> <file2> ...</code> | When you specify the <code>-i</code> or <code>-p</code> 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. |
| <code>[path]</code> | 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

| | |
|----------------|-----------------------|
| <code>s</code> | 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.

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-Exiting 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 |

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. Thereconnect 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> ...
```


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"
```

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
- <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.

There are three disk statuses:

- OK - When the system is up and running either as dual or single.
- Cleared - When you have cleared the database of a failed server.
- Unknown - When you reconnect the CPUs following a diskclear. When the diskcopy procedure has completed and the database has been mirrored, the disk status will change back to OK.

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

A message similar to the following appears:

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

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

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

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] <queue name>
```

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

The -i option allows you to ignore the inverted database trait.



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 |
| destination | mail |
| editmode | realname |
| keyboard | su |

utraits Command Options

| | |
|----------|----------|
| readrate | home |
| mail | password |



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 | ka | Kill all |
| c | Connect | log | Log client/server communication |
| cc | Configure colors | o | Order |
| cs | Configure shortcut toolbar | s | simplified user interface |
| er | Enter and remove | vb | Video browse |

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.

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 <queue name> [all]
```

Displays who locked a story.

For instance, to find out who locked a story in PEOPLE.SMITH, type:

```
wholockedit people.smith
```

To find out who last modified each story in a queue, type `whoLockedit` followed by the name of the queue and the keyword all.

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.

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.

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, sep, 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.

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.

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](#)
- [console.cfg](#)
- [SYSTEM.CLIENT.WINDOWS](#)
- [SYSTEM.CONFIGURE.301-ACTION](#)
- [SYSTEM.MAP](#)
- [SYSTEM.RESOURCE](#)
- [SYSTEM.WIRES.DISTRIBUTION](#)
- [SYSTEM.WIRES.KEYWORDS](#)
- [SYSTEM.WIRES.KEYWORDS-AP](#)
- [SYSTEM.WIRES.KEYWORDS-AP2](#)

/etc/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    nrca-a1 a1 nrca-a1.local
123.123.123.96    nrca-b1 b1 nrca-b1.local
10.1.1.1          nrca-a  a  nrca-a.local
10.1.1.2          nrca-b  b  nrca-b.local

# FTS
123.123.123.54    ftserver
```

/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

```

B System Files

```
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
```

B System Files

```
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      -      ;
```

B System Files

```
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      -      ;
```

B System Files

```
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 | - | - | ; |

B System Files

| | | | | | |
|--------|-----|-------|---|---|---|
| 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
```

B System Files

```
;  
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  purgelimt=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                purgelimt=0

; localtimeout=0:00     readrate=180
```

console.cfg

```
log b:log 32764

computer          ; A server
name              a
label             NRCS
irq               3
hostess           2c0
speed             9600
;
computer          ; B server
name              b
label             NRCS
irq               3
hostess           2c8
speed             9600
;
computer          ; C server
name              c
label             NRCS
irq               3
hostess           2d0
speed             9600
;
modem              ; remote console
password           issecret
timeout           6:00
```

```
irq      3  
hostess  2d8  
speed    2400
```

SYSTEM.CLIENT.WINDOWS



Either IP address or network card information is acceptable; IP address is preferable.

```
125.1.100.1 ;02608cdb7a2 ns001 big table
125.1.100.2 ;02608cd95e7e ns002 brock 07
125.1.100.3 ;02608c7e178e ns003 nydam 16
125.1.100.4 ;02608c7e67aa ns004 lockett 38
02608c7e519f ;ns005 michel 87
02608c7e1790 ;ns006 thibault 22
02608c7e51a8 ;ns007 ries 04
02608c7e6c01 ;ns008 lucas 57
125.1.100.9 ;02608c7e52c6 ns009 christensen 48
125.1.100.10 ;02608c7e5260 ns010 betty 69
125.1.100.11 ;0020aff431ff ns011 aveson 28
125.1.100.12 ;02608c7e6bfc ns012 robinson 49
125.1.100.13 ;02608c7e5274 ns013 tinsley 63
125.1.100.14 ;02608c7e6b58 ns014 reed 44
125.1.100.15 ;02608c7e532e ns015 landis 61
125.1.100.16 ;0020aff42efc ns016 dorr 51
125.1.100.17 ;0020aff42dad ns017 donnallen 60
125.1.100.18 ;0020aff42dcb ns018 douda 54
125.1.100.19 ;0020aff432ca ns019 kennedy 52
125.1.100.20 ;0020aff42d42 ns020 control room
125.1.100.21 ;0020aff431bf ns021 becker 30
125.1.100.22 ;0020aff43277 ns022 glass 29
```

SYSTEM.CONFIGURE.301-ACTION

```
scan      system.shredder
          remove

scan      system.cables.master
          dup   system.cables.groups
          dup   system.cables.cable#
          dup   system.cables.device_type

scan      phones.airports
          dup   phones.*all

scan      phones.business
          dup   phones.*all

scan      phones.fire
          dup   phones.*all

scan      phones.government.federal
          dup   phones.*all

scan      phones.government.local
          dup   phones.*all

scan      phones.government.state
          dup   phones.*all

scan      phones.hospitals
          dup   phones.*all

scan      phones.misc
          dup   phones.*all
```

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> <tpl
drive>:<tpl dir><start address> <end address> <address offset>
;  mos <device name> -
;
;=====
; 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.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 |

| | | | | | |
|----|-----------|------|---|---|-----------------------------|
| 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 |
| | | | | | ; |

B System Files

```
; 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.medsources |
| 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 |

B System Files

| | | | |
|-----------|-------------------------|--------|--------|
| 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)

B System Files

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 |
|-----------------------------------|-------------------------------|
| <code>/site/dict/messages</code> | Utility Messages |
| <code>/site/dict/queues</code> | Required Queues |
| <code>/site/dict/words</code> | Status and Option Words |
| <code>/site/dict/smessages</code> | Connect Commands and Messages |
| <code>/site/dict/keymacros</code> | Key Names for Macros |
| <code>/site/dict/shift</code> | Case-Shifting Parameters |
| <code>/site/dict/mcs</code> | 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 |
|----------------------|---------------------------|
| <code>W_ERROR</code> | <code>/ERROR</code> |
| <code>W_DEST</code> | <code>/destination</code> |

| Standard Name | Translation |
|---------------|-------------|
| 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. Select all servers at the console, 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)
```

C Standard Dictionaries

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](#)

C Standard Dictionaries

- [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_NOTQUEUE | /Not a queue |

| Standard Name (Continued) | Translation |
|----------------------------------|------------------------------------|
| M_PURGEZERO | /Queue is never purged |
| M_SYSEERROR | /System error |
| M_CATLONG | /Category code word too long |
| 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 |

| Standard Name (Continued) | Translation |
|---------------------------|--|
| M_KEYLONG | /Keyboard description contains too many characters |
| M_KEYREP | /Warning: a key definition contains a repeating function |
| 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 #0:d: |
| M_NOLOCATE | /could not locate "%c%d" |
| M_BADMEMORY | /memory allocation error |
| M_REFERENCE | /circular reference to macro #0: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_IGNOREING | /ignoring: %s |
| M_UNEXPECTED | /unexpected: %s |

| Standard Name (Continued) | Translation |
|----------------------------------|--|
| M_TWODEFS | /duplicate macro definition: |
| M_LONESTATE | /isolated keyboard state: |
| M_DISTRIBUTE | /%s does not distribute |
| 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_STDSRIPT | /Warning: "Script Swap" key redefined: |
| M_STDPRRIORITYQUEUE | /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" |

| Standard Name (Continued) | Translation |
|----------------------------------|---|
| M_GRPBADGRP | /Invalid name follows word "group" |
| M_GRPBADDEV | /Not a workstation device |
| M_GRPERRSKIP | /Group or alias word missing. Skipping text |
| M_GRPExINPUT | /Ignoring words following group name |
| M_GRPExINPUT | /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_GRPPIX | /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 |

| Standard Name (Continued) | Translation |
|---------------------------|---|
| M_GRPSOSO | /GROUPS story accepted, with errors |
| M_GRPNOUSER | /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 | /HOST-CCU COMMUNICATION ERROR |
| M_WIREIDLE | /Iwire has been idle for |
| M_WIRERESUME | /Iwire 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 |

Validation (Action) Server

| Standard Name | Translation |
|---------------|---------------------------|
| M_VALID | /Story valid |
| 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 |
| 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 | /!Group, 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.

C Standard Dictionaries

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 |
|--|--|
| <code>Q_ACCT</code> | <code>/system.account</code> |
| <code>Q_CATWORDS</code> | <code>/system.wires.distribution</code> |
| <code>Q_CLIENT_API</code> | <code>/system.client.api</code> |
| <code>Q_CLIENT_API_VERSIONS</code> | <code>/system.client.api-versions</code> |
| <code>Q_CLIENT_COM</code> | <code>/system.client.com</code> |
| <code>Q_CLIENT_COMMUNITY_VERSIONS</code> | <code>/system.client.community-versions</code> |
| <code>Q_CLIENT_INSTINCT</code> | <code>/system.client.instinct</code> |
| <code>Q_CLIENT_INTERPLAY_INSTINCT</code> | <code>/system.client.interplay-instinct</code> |
| <code>Q_CLIENT_VERSIONS</code> | <code>/system.client.versions</code> |
| <code>Q_CLIENT_WEB</code> | <code>/system.client.web</code> |
| <code>Q_CLIENT_WEB_VERSIONS</code> | <code>/system.client.web-versions</code> |
| <code>Q_CLIENT_WINDOWS</code> | <code>/system.client.windows</code> |
| <code>Q_CLIENT_WIRE_SERVER</code> | <code>/system.client.wire-server</code> |
| <code>Q_COLORS</code> | <code>/system.colors</code> |
| <code>Q_CONFIGURE</code> | <code>/system.configure</code> |
| <code>Q_DEAD</code> | <code>/dead</code> |
| <code>Q_DESTINATION</code> | <code>/notes</code> |
| <code>Q_FLASH</code> | <code>/wires.advisory.priority</code> |
| <code>Q_FORMS</code> | <code>/system.forms</code> |
| <code>Q_FTS</code> | <code>/system.fts</code> |
| <code>Q_GROUPS</code> | <code>/system.groups</code> |
| <code>Q_HELP</code> | <code>/system.help.terminal</code> |

| Standard Name (Continued) | Translation |
|----------------------------------|--------------------------|
| 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_SCRIPT | /system.dialogs |
| Q_SCRIPT_TEMPLATES | /system.script-templates |
| Q_SEARCHTAPE | /system.searchtape |
| Q_SEEK | /system.seek |
| 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 |

| Standard Name (Continued) | Translation |
|---------------------------|------------------|
| 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 | /FLASH |
| W_BULLETIN | /BULLETIN |
| W_URGENT | /URGENT |
| W_SILENT | /SILENT |
| W_ALWAYS | /A |
| W_TRANSMIT | /TRANSMIT |
| W_WIRE_FORM | /wires |
| Status Types | |
| W_MAIL | /mail |
| W_HOLD | /HOLD |
| W_LOCKED | /LOCKED |
| W_READY | /READY |
| W_NOTREADY | /? |
| W_NEW | /NEW |
| W_WIRE | /WIRE |

| Standard Name (Continued) | Translation |
|----------------------------------|--------------------|
|----------------------------------|--------------------|

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 |

You can change these translations only once. If you make a mistake, or want to change them again, you must extract the news program from the release CD first. Call iNews Customer Support for assistance in extracting the program.

| | |
|--------|-------|
| W_PAGE | /page |
|--------|-------|

Words Relating to the Seek Server

| | |
|--------------|----------|
| W_FAST | /fast |
| W_INDEXED | /INDEXED |
| W_ACTIVE | /ACTIVE |
| W_DONE | /DONE |
| W_ERROR | |
| W_ABORT | /ABORTED |
| W_DELIMITERS | .,' |
| W_PENDING | /PENDING |

| Standard Name (Continued) | Translation |
|-------------------------------------|---|
| W_SLOW | /slow |
| Words relating to FTS Server | |
| W_INDEXBASE | /C:/FTS |
| W_BINDFTSI | /ftsserver:6100 |
| W_BINDFTSS | ftsserver:6101 |
| Miscellaneous Options | |
| 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 |
| 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 |
| 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 |

C Standard Dictionaries

| Standard Name (Continued) | Translation |
|---------------------------|-------------|
| 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 |
| 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)

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.

C Standard Dictionaries

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 |

C Standard Dictionaries

| 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 stylelist |
| 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_BADMAPQTIME | /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_BADMAPNOMOS | /Bad Map line: MOS device not assigned to MOS Server |
| 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 |
| A_INVLADD | /Invalid address |
| A_INVLARG | /Missing argument |
| A_XMAXADD | /Exceeded address range |
| A_NOFORM | /Cannot access form |
| A_SRTERR | /Error sorting queue |

C Standard Dictionaries

| Standard Name (Continued) | Translation |
|----------------------------------|---|
| 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 |

| Standard Name (Continued) | Translation |
|---------------------------|-------------|
| 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_EVINCMPLT | /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 |
| bpoll | /bpoll |
| bscan | /bscan |
| charset | /charset |
| duplicate | /du |
| eof | /eof |
| every | /every |
| everyentry | /everyentry |
| extension | /extension |
| fast | /fast |
| ignore | /ignore |

C Standard Dictionaries

| Standard Name | Translation |
|---------------|---------------|
| ignore-del | /ignore-del |
| mailto | /mailto |
| move | /mov |
| number | /number |
| on | /on |
| open | /open |
| order | /or |
| 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 |
| 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 |

| Standard Name (Continued) | Translation |
|----------------------------------|--------------------|
| 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/smessages)

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 |
| S_CHEOL | /heol |
| S_CTYPE | /type |
| S_CWAIT | /wait |
| S_CMESS | /message |
| S_CEXPECT | /expect |
| S_CDELAY | /delay |
| S_CTIMER | /timer |
| S_CPASS | /pass |

C Standard Dictionaries

| Standard Name (Continued) | Translation |
|----------------------------------|--|
| 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 |
| 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 |

| Standard Name (Continued) | Translation |
|----------------------------------|---|
| S_ESCHLP | / <code><c></code> 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](#)

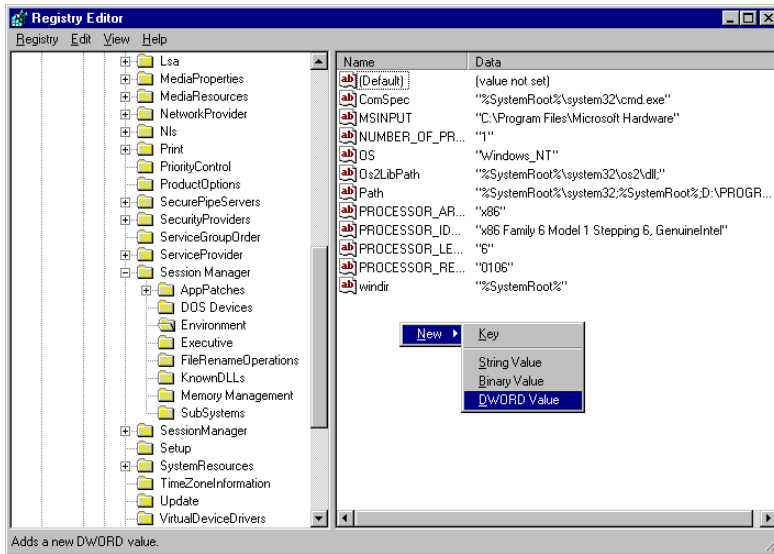
Registry Editor

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.

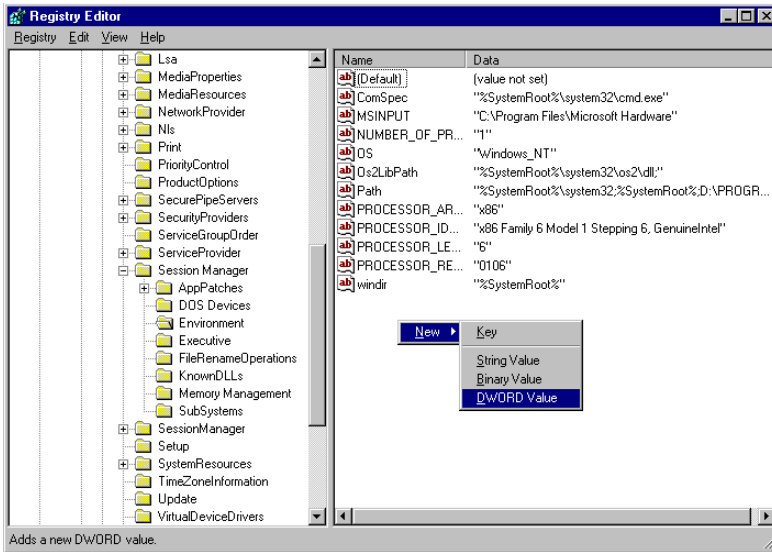
CCColor

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:

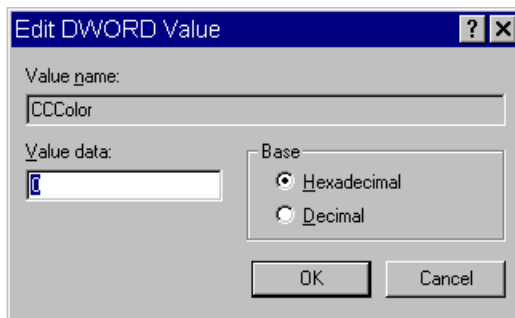
1. Open Registry Editor. See [“Registry Editor” on page 604](#) for more information.
2. Navigate to the Environment key and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu appears.

D Environment Variables



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 611](#) for more information on possible colors used in this environment variable

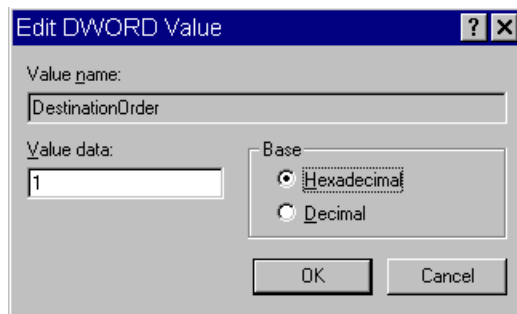
DestinationOrder

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 604](#) 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.



D Environment Variables

- 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.

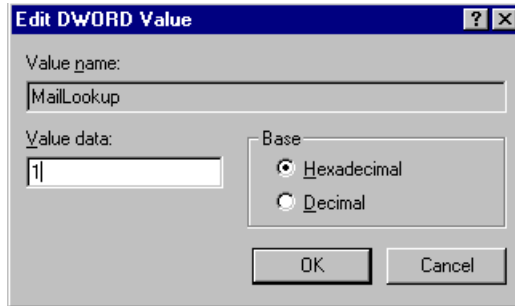


This environment variable must be created and defined in the Registry Editor at each workstation. Default behavior is used at workstations where the environment variable is not defined.

To set the environment variable and hide groups from e-mail lists:

1. Open Registry Editor. See [“Registry Editor” on page 604](#) 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.

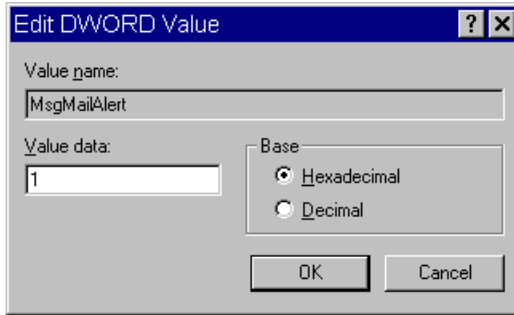
To set the environment variable thereby enabling message mail alerts:

1. Open Registry Editor. See [“Registry Editor” on page 604](#) 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.

D Environment Variables

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.
 - d. Click OK to save the setting and close the dialog box.
8. Close the Registry Editor window.

PIColor

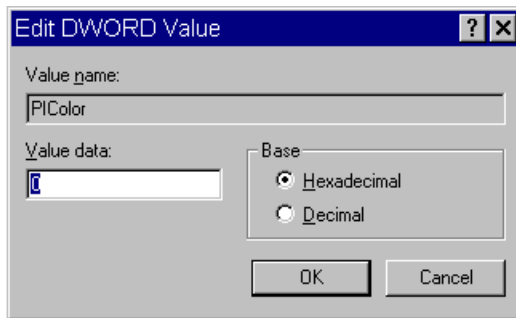
An individual iNEWS Workstation can have its presenter instructions text color changed via an environment variable called PIColor. If no environment variable exists, then the default color of red is used.

To change presenter instructions text color:

1. Open Registry Editor. See [“Registry Editor” on page 604](#) 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 | |

D Environment Variables

| 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 | |

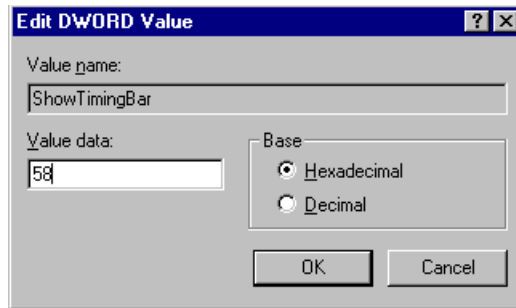
ShowTimingBar

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 604](#) 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 |

D Environment Variables

| 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 |

D Environment Variables

| 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.

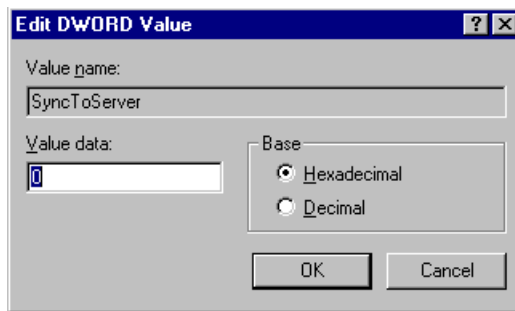


The synchronized timing feature should be enabled only at those workstations used to time a show. If the SyncToServer registry value is not created and defined at a workstation, then the synchronized timing feature is disabled at that workstation.

To enable the synchronized timing feature:

1. Open Registry Editor. See “Registry Editor” on page 604 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

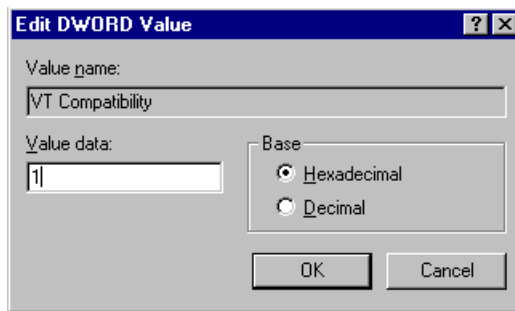
System administrators can set a limit for text in story form fields. For this feature to work, a Registry value defined as VT Compatibility must be added in the Environment key of the Registry at each workstation. If a registry value is not found in the Environment key of the Registry, then the default behavior—no text limit—is observed at the workstation.

D Environment Variables

To create and define this value:

1. Open Registry Editor. See “Registry Editor” on page 604 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

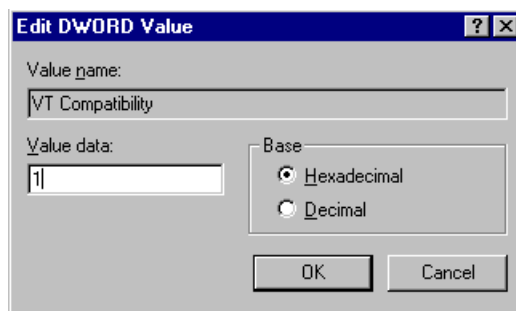
The environment variable called DisableCommandLine lets you disable the Command Line option in the Customize Toolbar Button dialog box, thereby restricting users from running executable programs within iNEWS on the PC. DisableCommandLine also disables any existing toolbar buttons that utilize the Command 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 604 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 Environment Variables

- 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 allows you to 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.

```
RXSITELISTSZ=<size>
```

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.

```
RXSITEFORMAT=<format>
```

The format controls which transfer protocol is initially used. This can be set to NSML or SEP. The default is NSML. This is how you get a foreign application, such as CNN which operates in SEP but did not use the FTP SITE command to change the format to SEP.

```
RXSEPFIELDLIST=<filename>
```

`RXSEPFIELDLIST` defines a file name. That file is processed as a list of field substitutions. For instance, `RXSEPFIELDLIST=/site/env/substitutes` in the `/site/env/rxnet` file, rxnet inherits this variable when invoked.

The format of each line of the field substitution file is: `<field identifier> = <field name>` where `<field identifier>` is a single character SEP field and `<field name>` is a valid NSML field name. An example would be `1 = video-id`. This causes the contents of the APP1 field, identified by 1, to be put into the NSML field video-id instead of the usual APP1-1.

```
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`.



You can also define an idle timeout value to override the "localtimeout" setting by inserting `RXSITEIDLE=<seconds>` in the file `/site/env/nxserver`.

```
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`.



The bold entires are exactly what to place in the file. Do not use an export statement after the line.

D Environment Variables

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  SOEKCVH  sc  queues
danielmi  180 0   n   i   -OEKCVH  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 `SOEKCVH` and `sc` are explained in the [“User Traits Console Command Summary” on page 632](#). See also [“list u” on page 509](#).
- 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 632](#) 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#
```

E Managing Traits at the Console

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 console 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 console 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.

E Managing Traits at the Console

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.
```

E Managing Traits at the Console

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 DDMMYY format. You must enter the days in double-digit format, meaning you must add a leading zero to single digit days, such as 05. Also, you must enter months as they are defined in the Words dictionary.

You can also specify a date range. This way, you can list users who changed their passwords sometime between two specific dates (date1 and date 2). For instance, to see if anyone changed their password after August 1, 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.

E Managing Traits at the Console

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 | <code>su n su -</code> | The user has superuser privileges. |
| Blacklisted | <code>+b -b</code> | The user cannot log in to the iNEWS system. |
| Simplified | <code>+s -s</code> | The user has limitations, as defined by the Simplified User Settings. Examples: <code>utraits palmer su n</code> <code>utraits palmer su -</code> <code>utraits palmer +b</code> <code>utraits palmer +s</code> |
| Local Only | <code>localonly</code> | The user can only log in locally. Example: <code>utraits jones +localonly</code> |
| Overwrite | <code>editmode o</code> | Everything the user enters in a story overwrites the character under the cursor. This is the default. Example: <code>utraits hansen editmode o</code> |

| Name in Modify User Account dialog box | Utrait Console Command | Definition / Example (Continued) |
|--|--------------------------|---|
| Insert | <code>editmode i</code> | <p>Everything the user enters in a story is inserted at the cursor location.</p> <p>Example:</p> <pre>utraits hansen editmode i</pre> |
| Home | <code>home</code> | <p>Sets the user's home directory, which usually contains the user's Mail and Notes queues.</p> <p>Example:</p> <pre>utraits loyd home people.1.loyd</pre> |
| Destination | <code>destination</code> | <p>Specifies the user's destination, which is usually a queue he or she uses frequently, such as the Notes queue.</p> <p>Example:</p> <pre>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.</p> <p>The system uses the readrate of the designated presenter to determine the audio (air) time of a story.</p> <p>Example:</p> <pre>utraits vanderberg 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> |

E Managing Traits at the Console

| Name in Modify User Account dialog box | Utrait Console Command | Definition / Example (Continued) |
|---|-------------------------------|---|
| Connect Services | +c -c | Allows user to connect to any service defined in the system. Example: <pre>utraits smith +c</pre> |
| Manage Projects | +mp -mp | Enables user to manage projects and facets in the system. Example: <pre>utraits jdoe +mp</pre> |
| Toolbars | +cs -cs | Allows users to create custom toolbars. Example: <pre>utraits carver +cs</pre> |
| Color Highlights | +cc -cc | Allows users to configure custom status colors. Example: <pre>utraits bagley +cc</pre> |
| Highlight Read Stories | +hr -hr | Specifies that unread stories in the queue are highlighted on the user's screen. Example: <pre>utraits davis +hr</pre> |
| Reorder Stories | +o -o | Specifies that user can create and remove stories from a folder or queue. Example: <pre>utraits lumpkin +o</pre> |
| Create/Kill Folders/Queues | +er -er | Specifies that user can create and remove entire folders and queues. Example: <pre>utraits delany +er</pre> |
| Kill All Stories | +ka -ka | Specifies that user can select an entire queue and move stories in it to the Dead queue. Example: <pre>utraits richards +ka</pre> |

| Name in Modify User Account dialog box | Utrait Console Command | Definition / Example (Continued) |
|---|------------------------|--|
| Password button | <code>password</code> | Provides capability to change the user's password. Example: <code>utraits jordan pass changeme</code> |
| Force Change check box | <code>force</code> | 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> |
| User Preferences button; Sessions tab; Keyboard field | <code>key ###</code> | Assigns a default keyboard (set of macros) to the user account. Example: <code>utraits vandenberg key 048</code> |

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:

```
SRPlo-LIsUGQsXWFiT sortfield purge dis mbox directory
-R-----I---Q-X----- TITLE      P4.0  D1  -    WIRES.ALL
```

A trait listing begins with a header line containing the letters SRPlo-LIsUGQsXWFiT; 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.

E Managing Traits at the Console

When one of these traits is on, the letter representing that trait appears in the second line. For instance, the R in the second line of the example indicates the Read-only trait is on:

```
SRPlo-LISUGQXWFiT 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-LISUGQXWFiT 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-LISUGQXWFiT 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-LIsUGQSXWFiT) 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.

E Managing Traits at the Console

| Name | Location in Display SRPlo-LIsUGQsXWFiT | Mode / Option Keyword | Definition / Example |
|-------------|---|--------------------------|--|
| Sequential | SRPlo-LIsUGQsXWFiT 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> |
| Read Access | SRPlo-LIsUGQsXWFiT -R----- | +r -r | Indicates whether or not stories in the queue are in read-only mode. To set the SHOW.5PM.SCRIPTS queue to read-only mode, type: <code>dbtraits show.5pm.scripts +r</code> 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. Turn Read Access on for queues in which people are likely to read but not change the stories. |
| Printable | SRPlo-LIsUGQsXWFiT --P----- | +p -p | Indicates whether you can use the print command on all stories in the queue with a single command when system printing. To enable users to print all stories in the SHOW.5PM.SCRIPTS queue with a single command: <code>dbtraits show.5pm.scripts +p</code> 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. |

| Name | Location in Display SRPlO-LIsUGQsXWFiT | Mode / Option Keyword | Definition / Example |
|---------------------------|---|--------------------------|--|
| Queue Being Ordered | SRPlO-LIsUGQsXWFiT ---lO----- | -o | <p>This is an indicator, rather than database traits. It indicates the queue's order status.</p> <p>The l 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-LIsUGQsXWFiT -----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-LIsUGQsXWFiT -----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> |

E Managing Traits at the Console

| Name | Location in Display SRPlo-LIsUGQsXWFiT | Mode / Option Keyword | Definition / Example |
|--------------------------|---|--------------------------|--|
| Sorted | SRPlo-LIsUGQsXWFiT -----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-LIsUGQsXWFiT -----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-LIsUGQsXWFiT -----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> |
| Queue Operations Allowed | SRPlo-LIsUGQsXWFiT -----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> |

| Name | Location in Display SRPlo-LIsUGQsXWFiT | Mode / Option Keyword | Definition / Example |
|-----------------|---|--------------------------|---|
| Save Version | SRPlo-LIsUGQsXWFiT -----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-LIsUGQsXWFiT -----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> |
| Watch Append | SRPlo-LIsUGQsXWFiT -----W----- | +w -w | <p>Assigned to queues that receive data from wire services. It allows the queue to monitor for new stories sent by the wire service, appends them to the wire queue, and immediately displays them to users who have that wire queue open. While this trait can be applied to any queue in 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:</p> <pre>dbtraits wires.all +w</pre> |

E Managing Traits at the Console

| Name | Location in Display SRPl0-LIsUGQsXWFiT | Mode / Option Keyword | Definition / Example |
|--------------------|---|--------------------------|--|
| Forms Allowed | SRPl0-LIsUGQsXWFiT -----F-- | +f -f | <p>Must be assigned to all queues in the forms directory. The forms will not work without this database trait applied. Additionally, this trait can be assigned to any queue in the database, but is usually only assigned to other queues that receive stories from other systems via rx/tx and then build forms for those stories, as needed. For instance, to assign the forms allowed trait to the SYSTEM.FORMS.R queue:</p> <pre>dbtraits system.forms.r +f</pre> |
| Index | SRPl0-LIsUGQsXWFiT -----i- | +index -index | <p>Assigned to queues and directories that you want to be indexed by the Fast Text Search (FTS) server. Allows for quicker searching of the queue or directory.</p> <p>For instance, to assign this trait to the WIRES.ALL queue:</p> <pre>dbtraits wires.all +index</pre> |
| Text Timing Clocks | SRPl0-LIsUGQsXWFiT -----T | +t -t | <p>Turns on the Text Timing Clocks, which appear in the Story Text panel. They are:</p> <p>TTC - Time to cursor BLK - Time of blocked (highlighted) text EST - Estimated read time of entire story</p> <p>To turn on these clocks in the RUNDOWN.5PM, type:</p> <pre>dbtraits rundown.5pm +t</pre> |
| Sortfield | SRPl0-LIsUGQsXWFiT 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> |

| Name | Location in Display SRPlo-LisUGQsXWFiT | Mode / Option Keyword | Definition / Example |
|-----------------------|---|--------------------------|---|
| Purge Interval | | purge | Indicates the “age” stories in a queue can reach before they are purged. See “Purge Interval” on page 29 for more information. |
| Display Lines/Refresh | | dis /+ -refresh | Indicates how many lines of each story in the queue are displayed. See “The dis Column” on page 33 for more information. |
| Queue Form | | qform | <p>The form used to display information in the Queue panel. Fields selected should be a sub-set of fields used in the story form. A field included in the queue form that does not actually exist in the story form cannot be written to in the Queue panel.</p> <p>In this example the rundown form is assigned as the queue form to the SHOWS.6PM.RUNDOWN queue:</p> <pre>dbtraits show.6pm.rundown qform rundown</pre> |
| Story Form | | sform | <p>Indicates the form assigned to be used in the Story Form panel (of the Story panel) to display form fields.</p> <p>In this example the futures form is assigned as the story form to the ASSIGNMENTS.TODAY queue:</p> <pre>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> |

E Managing Traits at the Console

| Name | Location in Display SRPlo-LisUGQSXWFiT | Mode / Option Keyword | Definition / Example |
|------------|---|--------------------------|---|
| Strip Form | | stripform | 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. |
| Mailbox | mbox | mail | Indicates mailbox assigned to the queue. See “Mailbox” on page 32 for more information. |
| 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> |
| Interplay | | interplay # | <p>Assigns an Interplay server to an individual directory. All directories default to zero (0).</p> <p>Here are a couple examples:</p> <pre>dbtraits show.sports interplay 1 dbtraits show.weather interplay 2</pre> <p>Use the <code>list d-i <directory></code> console command to see which Interplay server is associated with a specified directory.</p> |

| Name | Location in Display SRPlo-LIsUGQsXWFiT | Mode / Option Keyword | Definition / Example |
|-----------|---|--------------------------|--|
| 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-QsXWFiT 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.

E Managing Traits at the Console

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-LIsUGQsXWfiT 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-LIisUGQsXWFiT 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
```

E Managing Traits at the Console

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-LISUGQXWFiT 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.

E Managing Traits at the Console

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 rwnrg=<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 rwnrg=<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>
```

E Managing Traits at the Console

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 -
```

E Managing Traits at the Console

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:

```
....$ 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:

```
....$ 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="425 326 935 460">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> <pre data-bbox="425 730 992 841">27d - Deletes line 27. 30,35d - Deletes lines 30 through 35. 40,\$d - Deletes every line from 40 to the end of the file.</pre> <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> <pre data-bbox="425 1086 1213 1137">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.</pre> <p>To replace all occurrences on a line, add /g to the end of the command, such as:</p> <pre data-bbox="425 1194 682 1215">g/asws/s/asws/inws/g</pre> <pre data-bbox="425 1234 1196 1284">g/websession/d - Finds all occurrences of websession and deletes the lines containing it.</pre> <p>Each line affected is displayed after the editing command is applied.</p> |

| Command | Description & Examples (Continued) |
|-----------------------------|--|
| i | <p>The Insert command inserts one or more lines before the selected line. For instance:</p> <pre data-bbox="486 322 872 487"> /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><code>18m20</code> - Line 18 becomes line 20. Lines 19 and 20 become 18 and 19.</p> <p><code>1,5m\$</code> - 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: <code>10,20p</code>.</p> <p>More examples of the Print command can be found in “Specifying Lines to Edit” on page 656.</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="486 1277 951 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 [254, 542](#)
 /site/config [249, 543](#)
 editing [252](#)
 testing system changes [253](#)
 websession resource [447](#)
 wires [293](#)
 /site/dict/dmessages [598](#)
 /site/dict/joblist [597](#)
 /site/dict/keymacros [589](#)
 /site/dict/mcs [593](#)
 /site/dict/messages [575](#)
 /site/dict/queues [583](#)
 /site/dict/smessages [599](#)
 /site/dict/words [303, 586](#)
 /site/system [254, 555](#)
 changing [255](#)
 parameters [256, 257](#)
 /site/web/configuration [461](#)
 /site/wires [293](#)
 adding wire profile [294](#)

A

Absolute date, defined [630](#)
 Access restrictions [652](#)
 Account queue form [220](#)
 Accounts, viewing users [66](#)
 Action servers
 adding [333](#)
 configure rundown mirroring [340](#)
 examples of [332](#)
 field validation [335](#)
 ordered queues and command [325](#)
 polling commands [340](#)
 validation job list commands [338](#)
 Action servers, validation messages [581](#)
 ALL queue [466](#)

AMCPID [424](#)
 Apache Web server [446](#)
 at command [527](#)
 Avid
 online support [24](#)
 training services [26](#)
 Avid Data Receiver [292](#)

B

Backup files [245](#)
 Banner format options [286](#)
 Batch indexing [377](#)
 bottom console control command [478](#)
 bpoll command [527](#)
 broadcast command [482](#)
 Broadcast message [482](#)
 bscan command [527](#)

C

CAP events [424](#)
 Capture command [534](#)
 Category program messages [576](#)
 cccolor [605](#)
 CG styles [400](#)
 creating [402](#)
 CG templates [226](#)
 access editor [241](#)
 backgrounds [227](#)
 creating new [233](#)
 editing [229](#)
 font presets [239](#)
 CG Title Entry [223](#)
 configuration [226](#)
 overview [224](#)
 security [241](#)
 Character mapping [536](#)
 Command line, defined [43](#)

- Command prompt, defined 43
- Command sets for servers 323
- Commands
 - Avid personnel only 476
 - defining 570
- Community 419
 - configuring 420
 - removing a system 424
 - restrictions 423
 - troubleshooting 425
 - viewing sessions 423
- Components, licensing and limits 247
- Composite list queues 387
 - assign forms to 390
- computer commands 478
- Configuration file 249
 - adding wire service 293
 - editing 252
 - testing system changes 253
- Configuration file, example 543
- configure command 482
- connect command 483
- Console commands
 - control commands 43
 - server commands 42
 - types of 41
- Console commands, summary of utraits 632
- Console configuration file, sample of 52
- Console configuration keywords 52
- Console control commands 478
 - using 43
- Console history logs 49
- Console multiplexor 33
 - assigning mailboxes to a queue 647
 - configuration file 556
 - configuration file, sample of 52
 - console history 46
 - display 36
 - editing console.cfg file 55
 - enter superuser mode 37
 - exiting 40
 - function key assignment 44
 - keyboard 35
 - log in as system operator 37
 - logging out of remote console 51
 - managing group traits 650
 - managing traits 623

- printing history 479
- remote commands 51
- selecting servers 45
- setup overview 34
- starting the program 35
- troubleshooting 40
- unbusy stories and queues 147
- using remote console 50
- viewing configuration file 55
- viewing user traits 624
- zooming in on server 46

- Console server commands 482
- Console superuser 37
- console.cfg file 55, 556
- ControlAir fields and forms 218
- ctraits 422
- ctraits command 483

D

- D messages dictionary 598
- Data Receiver 292
 - adding wire profile 294
 - configuration and licensing 292
 - reconfiguring system 296
 - wire distribution 297
 - wire distribution story 295, 297
 - wire profile options 296
- Database components 354
- Databases 99–149
 - assigning queue forms 208
 - assigning story forms 208
 - changing traits 136
 - changing traits at console 637
 - creating a new directory 102
 - creating a new queue 104
 - creating a new story 111
 - deleting directories or queues 115
 - display lines trait 648
 - free up space 646
 - FTSindex attribute 649
 - getting info. on traits at console 635
 - group access 169
 - group traits, types of 170
 - Interplay attribute 650
 - locked queues and stories 141
 - mailbox attribute 647

- managing traits from console [623](#)
- overview [100](#)
- parent directory [637](#)
- purge interval trait [646](#)
- purge intervals and limits [139](#)
- refresh trait for queues [649](#)
- renaming a directory or queue [113](#)
- restricting access [652](#)
- restrictions [101](#)
- set up for projects [467](#)
- sortfield trait [645](#)
- summary of traits [122](#), [637](#)
- unbusy stories and queues [147](#)
- usage restrictions [169](#)
- viewing story information [118](#)
- viewing traits [116](#)
- date1 and date2 [630](#)
- dbclean command [484](#)
- dbclose command [484](#)
- dbdev command [485](#)
- dbdump command [485](#)
- dbfree command [486](#)
- dblines command [487](#)
- dbmanager [97](#)
- dboriginal command [488](#)
- dbpurge command [488](#)
- dbrestore command [489](#)
- dbserver command [491](#)
- DBServer program messages [576](#)
- dbsort command [491](#)
- Dbtraits
 - messages [583](#)
 - summary of console commands [637](#)
- dbtraits command [492](#)
- dbvisit command [493](#)
- Dead letter queue [106](#)
- Dead queue, recovering stories from [115](#)
- delay command [534](#)
- Destination queue [299](#), [345](#), [409](#)
- destinationorder [607](#)
- Devices
 - adding [265](#)
 - supported [262](#)
 - viewing information [263](#)
- diag command [534](#)
- Dialog boxes
 - Change User's Password [74](#)
 - Directory Properties [122–136](#)
 - Local Printing [274](#)
 - Manage User Accounts [66](#)
 - Modify User Account [69](#), [75](#)
 - Modify User Account, explained [632](#)
 - Preferences [76](#)
 - Queue Properties [122–136](#)
 - Simplified User Settings [83](#)
- Dialog commands [533](#)
- Dictionaries [569](#)
 - case-shifting [591](#)
 - ccucmds [570](#)
 - ccumsgs [570](#)
 - customizing [572](#)
 - d messages [598](#)
 - job list command [597](#)
 - keyboard macros [589](#)
 - mcs [593](#)
 - MCS files [397](#)
 - restoring defaults [574](#)
 - s messages [599](#)
 - update system [397](#)
 - utility messages [575](#)
 - words [586](#)
- dictionary command [495](#)
- Directories [99](#)
 - changing the parent [637](#)
 - creating a new [102](#)
 - deleting [115](#)
 - hiding [175](#)
 - hiding SYSTEM directory [652](#)
 - removing restrictions [653](#)
 - renaming [113](#)
 - restrictions [101](#)
 - viewing database traits [116](#)
 - Web access templates [450](#)
- DIRECTORY templates [122](#)
- Dis column [648](#)
- disablecommandline [619](#)
- Disconnect program messages [576](#)
- diskclear command [495](#)
- diskcopy command [496](#)
- Distribution codes, action servers and tx links [346](#)
- distribution command [527](#)
- Distribution servers [344](#)
 - adding [350](#)
 - codes [345](#)

- destination queues [345](#)
- dup and move commands [346](#)
- wildcards [345](#)

Distribution story [295](#), [297](#)

doc command [496](#)

down keyword console control command [478](#)

down number console control command [478](#)

dup command [346](#), [528](#)

E

Easy lock [145](#)

echo command [535](#)

ed command [496](#), [655](#)

- editing commands [659](#)
- quitting the line editor [662](#)
- saving changes [662](#)
- searching a file [658](#)
- specifying lines to edit [656](#)
- starting the line editor [656](#)

Edit lock [144](#)

Editing commands [659](#)

Editorial group [170](#), [173](#), [651](#)

enter command [497](#)

Environment variables [603](#)

- cccolor [605](#)
- destinationorder [607](#)
- disablecommandline [619](#)
- for servers [620](#)
- maillookup [608](#)
- msgmailalert [609](#)
- nxserver [620](#)
- picolor [610](#)
- registry values [605](#)
- RX_HOT_TO_GO [620](#)
- RXDEBUG= [620](#)
- RXDEBUGFILE [620](#)
- RXSITELISTSZ= [620](#)
- showtimingbar [612](#)
- synctoserver [616](#)
- VT compatibility [617](#)

Errors

- group checker [157](#)
- keyboard checker [189](#)

Escape character [535](#)

escape command [535](#)

Event list queues [387](#)

- assign forms to [390](#), [390](#)
- every command [528](#)
- expect command [535](#)
- extension command [528](#)

F

Facets [465](#), [471](#)

- list p command [506](#)
- overview [466](#)

fast command [528](#)

Fast text search

- archival and backup [379](#)
- batch indexing [377](#)
- components, installing [372](#)
- dynamic indexing [379](#)
- indexing stories [369](#)
- reindexing [379](#)
- servers [369](#)
- set up FTS components [374](#)

Field validation [335](#)

- using with servers [336](#)

Fields

- custom colors [207](#)
- customizing in forms [201](#)
- label borders [206](#)
- predefined lists [207](#)
- types of [208](#)

Find All form [221](#)

Flags [509](#)

Folders, creating a new [102](#)

Fonts, preset [239](#)

force command [497](#)

force console command [626](#)

Forms [197](#)

- assign to queue or story [208](#)
- creating [198](#)
- custom colors [207](#)
- customizing [201](#)
- field types [208](#)
- for ALL or QUERY queues in projects [468](#)
- for facets [471](#)
- guidelines [198](#)
- predefined lists [207](#)
- standard system forms [220](#)

FTS

- FTSindex attribute [649](#)

- index bases 649
- Ftsindex 374
- FTSindex attribute 649
- Ftsseek 374
- Ftsseek messages 582
- function key console control command 479
- Function keys 44
 - predefined system keys 186

G

- Group checker 156
- Group traits 170
- Groups 151
 - access and restrictions 169
 - add users as members 162
 - adding groups as members 163
 - adding workstations 166
 - avoiding recursion 163
 - changing multiple 161
 - choosing a name 154
 - create at console 154
 - create group memberships 155
 - creating 154
 - creating multiple 161
 - deleting 161
 - group checker messages 579
 - mail aliases 176
 - managing traits from console 650
 - membership possibilities 162
 - overview 152
 - permissions 167
 - renaming 160
 - restricted 174
 - timeouts 167
 - transferring assignments 174
 - view info. at console 152
 - view info. at workstation 153
- grpcheck command 497
- Grpcheck messages 579
- gtraits command 152, 498

H

- help command 498
- heol command 536
- history command

- bottom 478
- list 479
- print 480
- top 480
- up 480
- hogs command 499
- Hosts file 254
- HTML export template 432
 - sample story 441

I

- idiff command 499
- ignore command 528
- ignore-del command 529
- Index field, compatibility errors 126
- iNEWS
 - adding devices 265
 - backup files 245
 - changing database traits 136
 - command references 475
 - community 419
 - configuration file 249
 - console history logs 49
 - creating a new directory 102
 - creating a new queue 104
 - creating a new story 111
 - database overview 100
 - databases 99
 - defining commands 570
 - defining messages 570
 - deleting directories or queues 115
 - dictionaries 569
 - hardware overview 34
 - incorporating new configuration 253
 - licensing components 247
 - local print style story 288
 - network systems 407
 - projects and facets 465
 - recovering killed stories 115
 - remote console 50
 - renaming a directory or queue 113
 - set automatic update 407
 - shutting down the system 60
 - single-server mode 62
 - standard forms 220
 - starting the console program 35

- starting the system 58
- supported devices 262
- system configuration overview 244
- system messages 425
- system parameters 256
- system profile files 254
- system profile parameters 257
- update system dictionaries 397
- viewing database traits 116
- viewing device info. 263
- viewing story information 118
- Web publishing 429, 446
- wire distribution 297
- wire distribution story 295, 297
- wire profile options 296
- wires 291
- words dictionary 303

iNEWS.log 425

Internationalization 303

Interplay attribute 650

Intersystem messaging 267

- receiving 268

J

Job list command dictionary 597

Job list commands 527

- field validation 338
- poll and bpoll 340

Job list story 320

- adding a scan line 321

Job lists

- dup or move commands 347
- overlapping 342
- validate commands 347

K

Key lock 145

Keyboard checker 189

Keyboard checker, program messages 577

Keyboard macros 183

Keyboards 179

- creating 181
- defaults for users 193
- defined 180
- keyboard checker 189

- testing macros 192

Keyword checker

- defined 306
- messages 311

Keyword checker, program messages 576

Keyword limitations 306

Keyword servers 360

- adding 361
- database components 360
- defined 303

Killed stories

- recovering from Dead queue 115
- view who did it 119

L

Label borders 206

Last login messages 583

Line editor, ed 496, 655

- starting 656

Linux commands 477

list B command 502

list C command 503

list c command 504

list c message columns, defined 264

list command 500

list console control command 479, 479

list d command 504

list g command 505

list p command 506

list q command 508

- syntax 508

list s command 508

list sq command 509

list u command 509

list u-v command 624

local command 529

Local printing 273

- banner format options 286
- creating print styles 280
- defined 274
- dialog box 274
- sample style story 288
- style options 280

Locks

- removing 146
- types of 144

logclose console control command [479](#)
 logopen console control command [479](#)
 logout command [479](#), [511](#)

M

Macros

- creating [183](#)
- default user keyboards [193](#)
- defined [179](#)
- dictionaries [589](#)
- keyboard check program messages [578](#)
- keyboard checker [189](#)
- keys quick reference [185](#)
- notes of caution [189](#)
- plain text [187](#)
- predefined function keys [186](#)
- repeating [188](#)
- state keys [187](#)
- testing keyboards [192](#)
- types of [180](#)

Mail

- aliases for Internet [176](#)
- creating an alias [176](#)
- dead letter queue [106](#)
- disabling all users [380](#)
- enabling new users [91](#)
- network [380](#)
- outgoing mail queue [105](#)
- reserved for wires [303](#)
- servers [380](#)

Mail form [221](#)

Mail, server messages [581](#)

Mailbox tasks [325](#)

- defined [320](#)

Mailboxes

- assigning to a queue [328](#)
- types of [326](#)

Mailboxes, listing all queues (at console) [647](#)

maillookup [608](#)

mailto command [529](#)

makemontab command [511](#)

makeshift command [511](#)

maketab command [512](#)

Manager accounts

- creating Database Manager [97](#)
- creating User Manager [96](#)

Map command [536](#)

Map story [391](#)

mapin command [537](#)

mapout command [537](#)

MCS dictionary [593](#)

- files [397](#)

- status reports [596](#)

Memberships [162](#)

message command [537](#)

Messages

- broadcast [482](#)

- defining [570](#)

Messaging [267](#)

Mirroring rundowns [339](#)

modem console control command [479](#)

Monitor server

- configuration checklist [383](#)

- creating [384](#)

- defined [381](#)

- using [405](#)

Monitor server, mcs dictionary [593](#)

MOS events [424](#)

MOSID [424](#)

move command [346](#), [529](#)

msgclean command [513](#)

msgmailalert [609](#)

N

Notification group [172](#)

Notification priority [299](#)

- all wires [300](#)

Notify group [170](#), [652](#)

- view assignment [153](#)

NSML [435](#)

NSML format [407](#)

number command [529](#)

nxserver [620](#)

O

offline command [517](#)

on command [530](#)

online command [517](#)

Online support [24](#)

open command [530](#)

order command [325](#), [530](#)

Order lock 144
 Ordered queues 325
 Ordered trait 138
 otod command 517

P

Parallel wire servers 353
 adding 354
 pass command 537
 Passwords
 change user passwords at console 626
 changing a user's 74
 checking 632
 listing change by date 630–631
 listing last change 629
 system administration 38
 pause command 538
 Permissions, restricting 174
 picolor 610
 poll command 531
 Polling commands 340
 Polling issues 343
 Preferences, users 75
 Preset fonts 239
 Print banner format options 286
 print command 480
 Print server messages 583
 Print styles 280
 options 280
 PRINTER display, console 479
 Printers 273
 Priority queue 322
 Procedures
 access registry editor 604
 add a scan line 322
 add a seek server 367
 add an action server 333
 add entry in map story 392
 add groups to group 163
 add new user account 90
 add rx/tx links to iNEWS 410
 add server utility program 317
 add users to group 162
 add wire profile 294
 add wire service to config. file 293
 add workstation to group 166
 adding a new workstation 266
 adding a parallel wire server 355
 adding keyword servers 362
 assign command to function key 44
 assign default user keyboards 194
 assign editorial group 173
 assign notification group 172
 assign read group 171
 assign write group 172
 assigning queue or story forms 208
 assigning to a queue 328
 avoid hidden category errors 302
 change /site/system 255
 change a queue's sort field 645
 change command assigned to function key 44
 change database traits 136
 change default dictionary value 573
 change group name 160
 change order of transmit queue 410
 change presenter instructions text color 610
 change superuser password 39
 change system operator password 39
 change user preferences 76
 change user's password 75
 changing user passwords 626
 check for recursion 165
 choose existing font presets 240
 choose group name 154
 configure rundown mirroring 341
 configuring iNEWS for Community 420, 467
 connecting as remote console 50
 create a mail alias 176
 create a new directory 102
 create a new form 198
 create a new queue 104
 create a new story 111
 create and configure monitor 384
 create CG style 402
 create composite queue 387
 create Database Manager account 97
 create event list queue 387
 create facets 471
 create group at console 154
 create group memberships 155
 create keyboards/macro sets 181
 create macro with state keys 187
 create new CG templates 234

- create new directory [87](#)
- create new group [154](#)
- create new queue [88](#)
- create new users [86](#)
- create or change multiple groups [161](#)
- create print styles [280](#)
- create projects [468](#)
- create repeating macro [188](#)
- create user manager account [96](#)
- customize a form [201](#)
- define banner in print style [287](#)
- define key used to advance timing bar [612](#)
- define user account template [89](#)
- define value for VT compatibility [618](#)
- delete directories or queues [115](#)
- delete function key assignment [44](#)
- delete group [161](#)
- deleting font presets [240](#)
- designate every entry queues [322](#)
- designate priority queue [322](#)
- disable command line option [619](#)
- edit /site/config [252](#)
- edit console.cfg file [55](#)
- edit dictionary file [398](#), [572](#)
- enable destination order [607](#)
- enable message mail alerts [609](#)
- enable new user to get mail [91](#)
- enable synchronized timing [617](#)
- enter console control commands [43](#)
- enter gtraits interactive mode [162](#)
- exit console program [478](#)
- exit superuser mode [38](#)
- exit the console [40](#)
- find who locked an unlocked queue [142](#)
- find who locked queue [141](#)
- find who locked story [142](#)
- find who order locked queue [142](#)
- hide groups from e-mail lists [608](#)
- incorporate configuration changes [253](#)
- install FTS components [372](#)
- list all groups [152](#)
- list all groups for a user [152](#)
- list all members of group [152](#)
- list story info. without Index field [120](#)
- list who duplicated a story [120](#)
- list who killed a story [120](#)
- list who moved a story [120](#)
- log in as system operator [37](#)
- log in to iNEWS via Web [447](#)
- log out all users [97](#)
- log out remote user from console [52](#)
- logging out of remote console [51](#)
- make backups of system files [245](#)
- modify user traits [68](#)
- modify wire distribution story [295](#)
- pause scroll of console history [47](#)
- prevent access to CG Title Entry [242](#)
- publish stories to Web [446](#)
- reconfigure system [296](#)
- recovering stories from Dead queue [115](#)
- reindex an index base [380](#)
- remove editorial group [173](#)
- remove notification group [173](#)
- remove ordered trait [139](#)
- remove queue's production lock [147](#)
- remove read group [171](#)
- remove story's edit lock [146](#)
- remove system from community [424](#)
- remove user account [95](#)
- remove write group [172](#)
- rename group [160](#)
- renaming directory or queue [113](#)
- reset server I/O ports [41](#)
- restart scroll at console [40](#)
- restart send and receive from server ports [41](#)
- restore standard translations [574](#)
- resume queue sorting function [139](#)
- resume scroll of console history [47](#)
- run site configuration test [253](#)
- saving preset fonts for CGs [239](#)
- search for user info. [92](#)
- selecting all servers [45](#)
- selecting one server [45](#)
- selecting two servers [45](#)
- send server commands to multiple servers [42](#)
- set automatic update [408](#)
- set up ftsseek and ftsindex [375](#)
- set up iNEWS database for projects [467](#)
- set up keyword server for wires [304](#)
- set up seek feature [368](#)
- set up simplified users [83](#)
- setting up Txnet [430](#)
- shutting down the system [61](#)
- start CG Template Editor [229](#)

- start ftsseek and ftsindex 376
 - start in single-server mode 63
 - start or restart queue sorting 646
 - start the console 35
 - starting ed, the line editor 656
 - starting the system 58
 - stop a server command 42
 - stop FTS from console 377
 - stop ftsseek and ftsindex 377
 - stop runaway program at console 41
 - take superuser privileges 38
 - test macros in keyboards 193
 - transfer group assignments 175
 - transmit stories with same form 407
 - unbusy stories and queues 147
 - unlock queue without key 147
 - unlock story without key 146
 - update iNEWS dictionaries 398
 - use field validation 336
 - using monitor server 405
 - view /site/config at console 253
 - view community session 423
 - view configuration file 245
 - view database info. from console 117
 - view database info. from workstation 117
 - view function key assignment 45
 - view group assignment 153
 - view group errors 157
 - view group memberships 153
 - view high/low watermarks 257
 - view licensing limits 248
 - view purge limit 257
 - view recent console history 48
 - view remote system in community 423
 - view story information 119
 - view system parameters in use 256
 - view system profile 246
 - view user accounts 66
 - viewing console.cfg 55
 - viewing device information 263
 - zoom in on one server 46
- Production lock 144
- Projects 465
- creating 468
 - list p command 506
 - overview 466
 - setting up iNEWS database 467
- publish command 531
 - Purge intervals 303
 - database limits 139
 - format information 646
 - put command 531
- Q**
- Q_UNKNOWN 574
 - QUERY queue 466
 - Queue templates 122
 - Queues 99
 - /site/dict/queues 583
 - assign forms to 390
 - assigning forms 208
 - change order of transmit queue 410
 - changing the sort field 645
 - configure rundown mirroring 340
 - create composite list 387
 - create event list 387
 - creating a new 104
 - dead letter queue 106
 - deleting 115
 - destination queue 299
 - dictionary 583
 - display preview lines in rundown 648
 - every entry queues 322
 - hiding 175
 - job lists 320
 - locking 141
 - mailbox attribute 647
 - mirroring rundowns 339
 - ordered via action servers 325
 - outgoing mail 105
 - priority and scan queues 322
 - refresh trait 649
 - removing locks 146
 - removing restrictions 653
 - renaming 113
 - restrictions 101
 - set up forms 390
 - turn off ordered trait in sorted queue 138
 - unbusy 147
 - update trait 409
 - viewing database traits 116
 - Web access templates 450
 - quiet command 531

R

- Read group 170, 171, 651
 - for projects 468
 - view assignment 153
- Read limitations 652
- reconnect command 518
- Recursion 163
- Registry editor 604
- Relative date, defined 630
- remote command 531
- Remote console 50
 - commands 51
 - logging out 51
- Remote systems 423
- remove command 519, 532
- rename command 519
- reorder command 520
- replace command 532
- Reserved mailboxes 326
- reset console control command 480
- restart command 520
- Restricted group 174
- RGB hexadecimal color chart 611
- rr kb su m SOEKCVRH sc 509
- Rule sets 307
 - removing 310
- Rundown mirroring 339
 - configuration 340
 - polling issues 343
- Rx links 407
 - adding 410
 - update trait 409
- RXDEBUG= 620
- RXDEBUGFILE 620
- RXSITELISTSZ= 620

S

- S messages dictionary 599
- Save error messages 583
- scan command 532
- Scan queue 322
- Script templates 112
- Searches
 - job lists 306
 - keyword story and rules 307

- wire keywords 303
- searchtape command 520
- Security
 - CG Title Entry 241
 - group access 169
 - hiding queues and directories 175
 - permissions 167
 - protecting SYSTEM directory 652
 - timeouts 167
- Seek server messages 582
- send command 521
- send-del command 532
- sendform command 533
- Server commands, using 42
- Servers 315
 - action servers 332
 - adding a distribution server 350
 - adding a seek server 367
 - adding a server program 316
 - adding action servers 333
 - adding keyword servers 361
 - adding parallel wire server 354
 - command sets 323
 - creating a monitor server 384
 - distribution codes 345
 - distribution servers 344
 - dup and move commands 346
 - environment variables 620
 - fast text search 369
 - field validation 335
 - job lists 320
 - Keyword servers 360
 - mail 380
 - mailbox tasks 325
 - monitor 381
 - monitor configuration checklist 383
 - ordered queues and command 325
 - parallel wire servers 353
 - print messages 583
 - processing deleted stories 324
 - seek servers 367
 - selecting one or more 45
 - system servers 367
 - types of 316
 - types of tasks 320
 - using monitor server 405
 - zooming in on one at console 46

- showtimingbar 612
- shutdown command 521
- Simplified user settings 83
- Single-server mode 62
- /site/dict/shift 591
- sitedump command 521
- siterestore command 522
- Sorted queues, turn off ordered trait 138
- Sortfield database trait 645
- source command 533
- Source queue 409
- SRPlo-LIsUGQSWFi, explained 637
- Standard mailboxes 326
- startup command 523
- State keys 187
- status command 523
- stop command 523, 538
- Stories 99
 - adding rx/tx links 410
 - creating a new 111
 - job lists 320
 - locking 141
 - recovering 115
 - removing locks 146
 - sending story forms 407
 - server processing deleted 324
 - set automatic update 407
 - transferring via Ethernet 407
 - unbusy 147
 - view who moved, duplicated or killed 119
 - viewing information about 118
 - Web publishing 446
- Story forms
 - assigning 208
 - compatibility errors 126
 - customizing 201
 - defined 126
- Styles
 - options for printing 280
 - sample style story 288
 - using 280
- su command 524
- Superuser
 - password 38
 - prompt 37
- Superuser mode 37
- synctoserver 616
- System administration passwords 38
- System commands 475
- System configuration 243
 - backup files 245
 - overview 244
- SYSTEM directory
 - restricting access 652
- System files 541
 - /etc/hosts 542
 - /site/config 543
 - /site/system 555
 - backing up 245
 - console.cfg 556
 - hosts file 254
 - system profile 254
 - SYSTEM.CLIENT.WINDOWS 558
 - SYSTEM.CONFIGURE 559
 - SYSTEM.MAP 560
 - SYSTEM.RESOURCE 562
 - SYSTEM.WIRES 565
 - SYSTEM.WIRES.KEYWORDS 566
 - AP sample 567
 - AP sample 2 568
 - viewing 245
- System function keys 186
- System operator
 - logging in 37
 - password 38
 - prompt 37
- System parameters 256
- System profile files 254
 - changing 255
- System profile parameters 256, 257
- System servers, types of 367
- SYSTEM.CLIENT.WINDOWS 558
- SYSTEM.COLORS 207
- SYSTEM.CONFIGURE 320, 559
- SYSTEM.FORMS 198
- SYSTEM.FTS 374
- SYSTEM.GROUP 497
 - change group name 160
- SYSTEM.GROUPS 156
- SYSTEM.INDEX 374
- SYSTEM.KEYBOARDS 181, 583
- SYSTEM.LISTS 207
- SYSTEM.MAIL.ERROR 106
- SYSTEM.MAIL.OUT 105

SYSTEM.MAP 560
 create an entry 391
 SYSTEM.MOS-MAP 424
 SYSTEM.PROPERTIES 123
 SYSTEM.RESOURCE 400, 562
 SYSTEM.SEEK 570
 SYSTEM.STYLES 280
 sample style story 288
 SYSTEM.WEBFORMS 432, 447
 SYSTEM.WIRES 295, 297
 SYSTEM.WIRES.DISTRIBUTION 565
 SYSTEM.WIRES.KEYWORDS 566
 AP sample 567
 AP sample 2 568

T

Tasks, types of 320
 Templates
 for queues and directories 122
 for scripts 112
 Timed-interval tasks 329
 defined 320
 examples of 330
 Timeout parameters 167
 timer command 538
 Tokens for FTS 374
 Toolbar macros 184
 Toolbars, disabling command line option 619
 top console control command 480
 Training services 26
 Traits
 changing database traits 136
 database summary 122
 turning off ordered trait 138
 viewing database traits 116
 viewing user 66
 Transmit queue 409
 change order 410
 Troubleshooting 24
 Tx links 407
 adding 410
 field validation 335
 ordered queues and command 325
 polling commands 340
 polling issues 343
 update trait 409

validation job list commands 338
 Txnet, setting up 430
 type command 539

U

umanager 96
 unbusy command 524
 up console control command 480
 Update trait 409
 Usage restrictions 169
 User lock 144
 User manager, creating account 96
 User traits 69
 Users 65
 accounts, viewing 66
 adding new user account 88
 changing passwords 626
 create new 86
 creating Database Manager account 97
 default keyboards 193
 enabling mail 91
 log out all 97
 managing traits from console 623
 modifying traits 68
 modifying traits from console 625
 password, changing 74
 preferences 75
 remove accounts 95
 searching for info. 91
 simplified settings 83
 summary of traits 632
 types of 70
 viewing traits at console 624
 Utility messages 575
 Utility programs 315
 adding 316
 fast text search servers 369
 field validation 335
 job lists 320
 mail servers 380
 monitor server 381
 monitor server configuration 383
 seek servers 367
 system servers 367
 using monitor server 405
 utraits command 524

V

validate command 338, 347, 533
Validation server messages 581
Validation, uses of 335
verbose command 636
version command 525
view console control command 481
VT compatibility 617

W

wait command 539
Web access 446
 configuration 461
 default directory template 456
 default queue template 458
 default story template 449
 directory and queue templates 450
 logging in 447
 story templates 447
Web publishing 429
 configuring 430
 HTML export template 432
 NSML conversion 435
 optional format strings 439
 sample HTML export template 441
Web server 446
Web sessions 447
wholockedit command 526
Wildcards 345
Wire distribution
 story instructions 347
Wire distribution, sample system file 565
Wire profile files
 defined 291
 options 296
Wire story form 222
Wires 291
 add wire service to config. file 293
 adding 292
 adding wire profile 294
 avoiding hidden categories 302
 backup distribution components 354
 configuration and licensing 292
 destination queue 299
 distribution 297

 expanded keyword servers 360
 internationalization 303
 keyword checker messages 311
 keyword searches 303
 keyword story 307
 multi-queue distribution 310
 notification priority 299
 parallel wire servers 353
 profile options 296
 program messages 581
 purge intervals 303
 reconfiguring system 296
 reserved mailboxes 303
 search job lists 306
 wire distribution story 295, 297

WIRES.ALL 300

WIRES.KEYWORDS 307

WIRES.UNKNOWN 301

Workstation, save error messages 583

Write group 170, 172, 651

 view assignment 153

Write limitations 652

X

x console control command 481

Z

zoom console control command 481



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