Alcatel·Lucent

Network Traffic Management

8920 Network Traffic Management software

Input Commands Guide
Release 17.3

190-406-811
Issue 1.0
October 2012

Notice

Every effort was made to ensure that the information in this document was complete and accurate at the time of printing. However, information is subject to change.

Trademarks

All trademarks and service marks specified herein are owned by their respective companies.

Warranty

Alcatel-Lucent provides a limited warranty to this product.

Customer Notification

The Alcatel-Lucent contract specifies your system configuration (e.g., capacities) and identifies the optional features you have purchased.
The standard NTM Feature Set documentation contains information on all of the features available in the Release, including those you may not have purchased, which are thereby not available for use.
Alcatel-Lucent will not support external use of the third-party software packages included in the NTM Feature Set.

Acknowledgements

We wish to acknowledge:

ACP50 and ACP550 are products of NetKit Solutions LLC.

The NTM product includes software developed by:

Red Hat Enterprise Linux® - Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

APACHE TOMCAT - The Apache License, version 2.0 (http://www.apache.org/licenses/).

APACHE ActiveMQ - The Apache License, version 2.0 (http://www.apache.org/licenses/).

MOD AJP (APACHE Tomcat Connectors) - The Apache License, version 2.0 (http://www.apache.org/licenses/).

Apache Xerces C++ - The Apache License, version 2.0 (http://www.apache.org/licenses/).

Apache Axis2 - The Apache License, version 2.0 (http://www.apache.org/licenses/).

Bugzilla - Mozilla Foundation; License: http://creativecommons.org/licenses/by-sa/2.0/

CentOS - CentOS Project;

Dom4J - DOM4J Project; License: http://www.dom4j.org/dom4j-1.6.1/license.html

LDAP C SDK - Mozilla Foundation; License: http://www.mozilla.org/MPL/MPL-1.1.html

mksh - Korn shell by David Korn; Distributed under BSD License. (https://www.mirbsd.org/htman/i386/man7/BSD-Licence.htm)

ncurses - ncurses, GNU 5.5; Distributed under MIT + GPL2+

nmon - IBM nmon; License: http://www.gnu.org/copyleft/gpl.html

PAM_RADIUS_AUTH - This module is a merger of an old version of pam_radius.c, and code which went into

mod_auth_radius.c, with further modifications by Alan DeKok of CRYPTOCard Inc.. The original pam_radius.c code is copyright (c) Cristian Gafton, 1996, redhat.com> The additional code is copyright (c) CRYPTOCard Inc, 1998. All rights reserved.

JAVA JDK - Sun Microsystems Inc. Binary Code License Agreement (http://java.sun.com/j2se/1.5.0/jdk-1_5_0_12-license.txt).

edtFTPj - Enterprise Distributed Technologies under LGPL License (http://www.gnu.org/licenses/lgpl.txt).

Perl DBD - Perl DBD Copyright (c) 1994-2003 Tim Bunce, Ireland is used with permission. Distributions of the standard package can be found through the http://www.cpan.org website.

Perl Convert::ASN1 - Perl DBD Copyright (c) 1994-2003 Tim Bunce, Ireland is used with permission. Distributions of the standard package can be found through the http://www.cpan.org website.

Perl URI - Perl DBD Copyright (c) 1994-2003 Tim Bunce, Ireland is used with permission. Distributions of the standard package can be found through the http://www.cpan.org website.

Prototype - Copyright (c) 2005-2007 Sam Stephenson

Scmbug - Scmbug by Martin Tomes; License: http://www.subversionary.org/projects/scmbug

SNMP4j - SNMP4J.org; License: http://www.snmp4j.org/LICENSE-2_0.txt

Subversion - CollabNet; License: http://subversion.tigris.org/license-1.html

SWISH-E - Copyright 1995-1998 by Miles O'Neal, Austin, TX, USA. GNU General Public License.

w4ais - Copyright 1995-1998 by Miles O'Neal, Austin, TX, USA. (http://yolo.net/w4ais/license.html)

GNU LESSER GENERAL PUBLIC LICENSE - Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy. This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to ecompile the application to use the modified definitions.)

b) Use a suitble shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances. It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

JSch 0.0.* was released under the GNU LGPL license. Later, we have switched over to a BSD-style license.

Copyright (c) 2002-2010 Atsuhiko Yamanaka, JCraft,Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors 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 ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contents

4 Control Commands

5 Record Base and Database Commands

6 Report Commands

7 System Administration Commands

8 UDDM/UDNEI Commands

9 Status and Other Commands

GL Glossary

IN Index

List of figures

1 Introduction to Input Commands

2 Commands for Auditing Network Elements

3 BDR Commands

4 Control Commands

5 Record Base and Database Commands

6 Report Commands

9 Status and Other Commands

List of tables

1 Introduction to Input Commands

2 Commands for Auditing Network Elements

3 BDR Commands

4 Control Commands

5 Record Base and Database Commands

6 Report Commands

7 System Administration Commands

8 UDDM/UDNEI Commands

9 Status and Other Commands

1 Introduction to Input Commands

Overview

Purpose

This chapter introduces the types of input commands available in 8920 Network Traffic Management software and describes their functions. It also contains general information needed for operating these input commands from a novice or an experienced user level.

Contents

This chapter contains the following topics:

□

Types of input commands

Overview

All input commands described in this document can be used regardless of switch type unless specifically noted otherwise.

Audit

Use audit commands to audit the switch and to inhibit or allow discrete-triggered audits.

Reference: Chapter 2, "Commands for Auditing Network Elements")

BDR commands

Use BDR (Backup and Disaster Recovery) commands to administer the BDR features, which are only available with BDR features (Feature 8, "Disaster Recovery (Duplex)" and Feature 40, "Enhanced Disaster Recovery").

Reference: Chapter 3, "BDR Commands"

Control

Use control commands to add, modify, and delete controls for a network office.

Reference: Chapter 4, "Control Commands"

Record base and database

Use record base and database commands to add, modify, delete, maintain, and access data in the 8920 NTM database.

Reference: Chapter 5, "Record Base and Database Commands"

Reports

Use report commands to modify, generate, or output reports. These commands are ***not*** related to Feature 272, "NTM Report Writer". For information on that feature, see the *Report Writer Guide*.

Reference: Chapter 6, "Report Commands"

System administration

Use system administration commands to set up and maintain the host machine, administer user accounts, etc.

Reference: Chapter 7, "System Administration Commands"

Status and other

Other commands include status commands (used to obtain the status of NTM or network elements), commands that allow you to send messages to other users, etc.

Reference: Chapter 9, “Status and Other Commands”

□

List of input commands

Table

Table 1 lists each NTM input command and its function.

Table 1 Input commands and functions (Sheet 1 of 6)

Command	Function
`acc`	Applies or removes an “automatic congestion control” command
`act`	Activates data collection for an office, switch, or DCC (Data Collection Concentrator)
`act_dcc`	Activates all the end offices defined for a DCC office
`addntmuser`	Adds new user IDs (Identifiers) to the *Linux* system “/etc/passwd” and “/etc/group” files
`audallow`	Enables discrete-triggered audits
`audinh`	Disables discrete-triggered audits
`audit`	Audits a switch or set of switches for any or all audit types
`audstat`	Obtains status of discrete-triggered audits
`bdr_act`	Activates Feature 8, “Disaster Recovery (Duplex)” and Feature 40, “Enhanced Disaster Recovery” on the NTM host computer
`bdr_audit`	Audits all backup entities
`bdr_chgstat`	Changes or checks the status of global record base files on the host computer
`bdr_commit`	Synchronizes changes made to the global record base files
`bdr_deact`	Deactivates Feature 8, “Disaster Recovery (Duplex)” and Feature 40, “Enhanced Disaster Recovery” on the NTM host computer
`bdr_switchbk`	Switches a host computer from takeover mode back to normal mode
`bdr_sync`	Synchronizes record base files on one host computer with those on another host computer

Table 1 Input commands and functions (Sheet 2 of 6)

Command	Function
`bdr_takeover`	Allows a remote host computer to take over the functionality of a failed host computer
`canf`	Applies, modifies, or removes a "cancel from" command
`cant`	Applies, modifies, or removes a "cancel to" command
`cg`	Applies, modifies, or removes a "call gap" command
`cr`	Applies, modifies, or deletes a "circuit reservation" control
`create`	Creates all or a portion of the record base
`cro`	Applies or removes a "cancel reroute overflow" control
`ctrlactlist`	Retrieves all active TG controls and Call Gap controls
`ctrlog`	Lists active or matched controls for all control types
`darpt`	Creates Detailed Analysis Report
`datastat`	Shows the data transmission status for NTM offices
`dayend`	Changes current database to next available database
`dbadmin`	Changes status of database
`dbstat`	Provides status of all Tuxedo databases
`dbtest`	Tests "/musr/rb" files for syntax errors and inconsistent data
`deact`	Deactivates data collection for the specified office(s) or DCC
`deact_dcc`	Deactivates all end offices associated with a DCC
`delntmuser`	Removes user IDs from the *Linux* system "/etc/passwd" and "/etc/group" files
`demand`	Retrieves specified data from the system database
`dhtr`	Applies or removes domestic HTR (Hard-To-Reach) codes
`dptpri`	Activates, modifies, or removes a "dynamic packet trunks prioritization" control
`dptres`	Activates, modifies, or removes a "dynamic packet trunks reservation" control

Table 1 Input commands and functions (Sheet 3 of 6)

Command	Function
`dpttid`	Activates, modifies, or removes a “dynamic packet trunks terminal identifiers” control
`dsc`	Monitors discretes
`encrypt_param`	Encrypts specific parameter in the instance switch file.
`errlog`	Views error log files
`exptrules`	Allows enabling/disabling of rules processing of exceptions
`failrep`	Outputs reports of switches with late measurement data or unsuccessful discrete collection
`firpt`	Creates Final Trunk Group Report
`fmltoasc`	Converts a fielded buffer to ASCII output
`guiwall`	Allows users to send out broadcast messages from the host machine
`histdata`	Changes from current mode to histdata mode
`htr_codes`	Lists the codes that belong to an office
`htr_ref`	Lists the offices that belong to a code
`icrpt`	Produces the Idle Capacity Report
`ihtr`	Applies or removes international HTR codes
`installdb`	Moves a newly built database into the current database area
`install_dcol`	Installs new UDNEI dcol.
`limitthr`	Limits the number of exceptions during a crisis.
`linkstat`	Displays data collection status of offices and DCC(s)
`local_audit`	Provides a quick approximation of an `audit all` on a switch
`longdemand`	Provides the capability to direct typically large processing and response time sql requests to a dedicated database server
`manage_audit`	Manages audits.

Table 1 Input commands and functions (Sheet 4 of 6)

Command	Function
`manage_control`	Manages controls.
`manage_ccrules`	Populates CC configuration tables.
`manage_dcol_params`	Manages data collection parameter files.
`manage_instance_params`	Manages parameter files.
`manage_uddm`	Manages UDDM data types.
`manage_udneitype`	Manages UDNEIType files.
`ncrpt`	Creates Daily No Circuit Report
`ofc_grep`	Searches “/musr/rb/rspte/rspte” for entries containing office set names
`ofcovrd`	Overrides a particular type of automatic control for a specific *5ESS* switch
`ofcset`	Retrieves and displays all offices from a given office set
`ongoing`	Retrieves specified data fields from the system's shared memory
`ongoing.test`	Verifies that the SQL (Structured Query Language) has no errors
`oprpt`	Creates Office Performance Report
`pas`	Displays public announcement service data for a specific *4ESS* switch office for the current 5-minute period
`perfrep`	Outputs performance reports based on historical data
`pp`	Applies or removes a preprogrammed trunk group control in a 1A *ESS* office
`pplist`	Lists all preprogram controls that can be activated or deactivated for a 1A *ESS* office
`preplan`	Edits or implements a control preplan
`purglog`	Removes matched control entries from the database

Table 1 Input commands and functions (Sheet 5 of 6)

Command	Function
`rbed`	Invokes the record base editor
`rc`	Applies routing code for call gapping (*GTD-5* only)
`rdbstat`	Provides status of all relational databases
`recreate`	Incorporates record base changes into the active database
`rr`	Applies, modifies, or deletes a "reroute" control
`sendmsg`	Verifies end-to-end connection between NTM and a DCC
`silc`	Applies or deletes a "selective incoming overload" control
`skip`	Applies, modifies, or deletes a "skip" control
`snw_admin`	Sets up and administers subnetworks
`snw_info`	Gives status information on subnetworks
`start_arc`	Starts an ARC process
`startnet`	Starts the Ethernet on the computer along with associated services
`startsys`	Starts or restarts the system
`stop_arc`	Stops an ARC process
`stopnet`	Gracefully shuts down the Ethernet on the computer
`stopsys`	Stops or shuts down the system
`sync_audit`	Synchronizes the data base controls that reside in the secondary subnetwork partitions on this NTM host
`sysstat`	Reports information about the current state of the system
`tg_grep`	Displays trunk group files within a given trunk group set
`tgset`	Displays trunk groups within a given trunk group set
`thrconvert`	Converts old (static) threshold or filter files into the new rules-based format
`thresh`	Creates a threshold table

Table 1 Input commands and functions (Sheet 6 of 6)

Command	Function
`total`	Removes all instances of a class of controls from an office
`udnei_act`	Start data collectors for specific UDNEIType.
`udnei_deact`	Stop data collectors for specific UDNEIType.
`urwformat`	Produces user report writer format
`viperm`	Command to safe editing of permissions file.

□

Entering input commands

Overview

You can use menus and mnemonics to enter input commands or you can enter input commands directly at the *Linux* system shell or at any **CMD:** prompt.

Input methods

There are three ways to enter input commands at the *Linux* system shell:

- ***Name-defined syntax:*** Enter all parameter phrases (parameter name, equal sign, and parameter value) on the command line. You can enter the parameters in any order.
- ***Position-defined syntax:*** Enter only the parameter values on the command line in position-defined syntax.

  Reference: "Name-defined and position-defined parameter syntax" (p. 14)
- ***Prompting:*** Do not enter all parameters on the command line. You will automatically be prompted for required parameters that were not entered on the command line.

  The following applies to the prompting of parameter values for each command:

  – Entering **RETURN** defaults a parameter.

  – Entering a semicolon (**;**) indicates you have entered your last parameter value.

  – Any optional parameters not specified use their specific default values.

Syntax

When using the NTM input commands at the *Linux* system shell, enter the input command name and parameters in the correct format as shown under the "Command syntax" (p. 12).

Table

Table 2 describes the notation standards for all NTM input commands.

Table 2 Notation used in this guide

Symbol	Function
Bold type	Indicates user-entered input
Italic type	Indicates user-entered variables
Square brackets [ ]	Encloses an optional parameter. For control commands, a parameter is shown as optional only if it is not required for all available actions (e.g., add, modify, delete).
CR or **RETURN**	Indicates carriage return or Enter
Semicolon (;)	Terminates prompting for all parameters

Table 2 Notation used in this guide

Symbol	Function
<blank>	Separates a command from its parameters
Comma (,) or <blank>	Separates parameters from other parameters
Question mark (?)	Help
exit	Terminates working session
DEL or **CTRL - d**	Aborts a command
Plus sign (+)	Separates multiple parameter values
Backslash (\ )	Separates multiple lines in preplans
Date/time in the form YY/MM/DD-HH:mm	Indicates date and time in this format: • YY = numeric year (last 2 digits of the year) • MM = numeric month (1-12) • DD = numeric day (1-31) • HH = hour (0-23) • mm = minute (0-59)

Entering numbers

When numbers are entered with NTM input commands, the following are the allowable ranges for:

- integers: 0 - 2,147,483,647, inclusive
- floating-point real numbers: 0.0 - 2,147,483,647.9, inclusive

Entering office names

Office names specified for various input commands can be one of the following:

- All (all offices)
- Network element name (such as “ess5”)
- Nickname for the office (such as clmb for clmbohxx11a)
- One or more of the initial characters of the office identifier (the highest ranking office in the RSPTE file matching those characters is selected)

  Reference: For more information on office rankings within RSPTE files, see the “RSPTE File” (p. 68) in the *Record Base Administration Guide*.

Understanding input command entries

Each input command entry consists of up to eight sections. Table 3 describes each of these sections.

Notes:

1. Some of the sections described in Table 3 do not apply to all input commands and will not appear with the entry.
2. Sections may not appear in the order specified below, in order to make the online HTML files more easily navigable.

Table 3 Input command sections in this guide

Section	Description
Command Name and Description	Describes the function of the input command. For example, the `audit` command is used to audit the switch for all audit types.
Syntax	Shows the proper format for entering the input command. The syntax consists of the command name followed by one or more parameter phrases. Each phrase contains a parameter name, an equal sign, and a syntax variable (see below).
Parameters	Defines any variables used in the syntax and lists the valid values for the corresponding parameter names. For example, assume that the *`action`* variable appears in the syntax of an input command under the "Syntax" entry. This section instructs you to enter `add`, `mod`, or `del` for the *`action`* variable.
System responses	Describes any system outputs generated by the syntax. The System responses section will only appear if the response is different than that shown in Figure 1.
Examples	Shows an example of the input command entry.
References	Lists supporting references. These references may include information about other input commands or NTM features.
Related commands	Lists commands related to the input command described. For example, the `startsys` command is used to start the NTM system. The `stopsys` command is a "related" command used to stop the NTM system.

Figure 1 shows the typical response for any command.

Figure 1 Typical system responses

IF the command is …	THEN the system responds with …
executed successfully	IP followed by the system prompt
not executed successfully	the appropriate error message

Command syntax

The message format for input commands is as follows:

`command_name parameters [<t>]`

`command_name` is the command itself (for example, `act`)

`parameters` is a set of zero or more arguments that specify how the command must be run

Important! Some parameters are mutually exclusive, based on values for other parameters.

`[<t>]` is the command terminator

- **RETURN** — The **RETURN** key runs the command, prompting for required parameters not already entered on the command line.

  Important! For the **?** and **!** terminators, you must enter a space between the command name and the terminator, which must be followed by **RETURN**.

- **?** — The question mark requests information about the command (e.g., syntax). Since the question mark (`?`) is a *Linux* pattern matching character, it matches any file in the current directory whose filename is one character long. If you have one-character filenames, you must precede the question mark with a backslash (`\?`) or enclose it in single quotation marks (`'?'`) to invoke the syntax help feature. Do this only on the command line, not when you are prompted for a parameter.
- **!** — The exclamation point prompts for all parameters that the user has not already specified. No default values will be provided. The user must enter a value for each displayed parameter name. If the user presses **RETURN**, the system responds with the default value if one exists. If no default exists, the user must fill in a valid value.

Getting help on syntax

If you are uncertain of proper syntax for a command, enter the command followed by a space, a question mark (`?`), and a **RETURN** to request the valid syntax of the command. Figure 2 shows a request for the valid syntax of the `act` command.

Figure 2 Request for valid syntax of act command

```
$ act ?
```

Result: The system responds with the proper syntax (Figure 3).

Figure 3 Proper syntax for act command

```
Usage:      act office = office+ ...+office , [type = type+
   ...+type] [<t>]
```

The `act` syntax indicates that the `act` command has the following parameters: `office` and `type`. The `office` parameter is required; it may have multiple values as indicated by the plus sign (+). The `type` parameter is optional (has a default). This parameter name is displayed as a prompt only when the exclamation point (`!`) is used as the command terminator.

Parameters

Overview

You can use one or more parameters with a command. Each parameter has two parts: a command name and a keyword with one or more values. Names and values can be shortened to three or more characters by truncation (for example, the name `xchnge` can be shortened to `xch`). For example, the `deact` command may be entered in any of the following three ways:

```
Example: deact office=clmb04 type=control
```

```
Example: deact off=clmb04 typ=con
```

```
Example: deact clmb04 con
```

You must separate the name from the value with an equal sign (=). Parameters are separated from each other with a space or a comma followed by a space. Figure 4 shows parameters separated from each other with spaces.

Figure 4 Parameters separated by spaces

```
$ act off=clmb04 typ=meas
```

Parameters containing multiple values

If a parameter allows multiple values, use the plus sign (+) as a separator between the values.

In cases where two parameters of a command both contain multiple values, both values from the second parameter are applied to each value of the first parameter. In Figure 5, both values of the parameter `type` (`meas` and `aud`) apply to each `office` value (clmb04 and clmb05).

Figure 5 Parameter with multiple values

```
$ act office=clmb04+clmb05 type=meas+aud
```

Name-defined and position-defined parameter syntax

When you enter commands at the *Linux* system shell, you can use either name-defined or position-defined syntax to specify each parameter.

Name-defined parameters allow you to enter parameters on the command line with all parameter names and parameter values specified. By entering each parameter name with its value(s), you can specify the parameters in any sequence you want.

For instance, the Syntax for the `ctrlog` command (as shown on page (4-59)) is:

```
ctrlog office=office type=type status=status [trap=trap]
   [by=login] [start=start] [stop=stop]
```

Figure 6 shows parameter entries for `ctrlog` in name-defined syntax.

Figure 6 Name-defined syntax command entry

```
$ ctrlog office=clmb04 type=cg+rr status=active by=all
```

In this example, the values `clmb04`, `cg+rr`, `active`, and `all` appear with their respective parameter names (`office`, `type`, `status`, and `by`). Although other parameters are available for the `ctrlog` command, if they are not required or are not supported by the specified office(s), the system will supply the default values or ignore them accordingly. If they are required, the system will prompt you with the parameter name and you will have to supply a valid value.

Position-defined parameters allow you to omit parameter names on the command line. However, values must be entered in the same order as shown in the heading "Syntax" for each command. Figure 7 shows parameter entries in position-defined syntax.

> **Important!** If you wish to omit a value in the middle of a list of position-defined parameters, use the comma (`,`) as a placeholder.

Figure 7 Position-defined syntax command entry

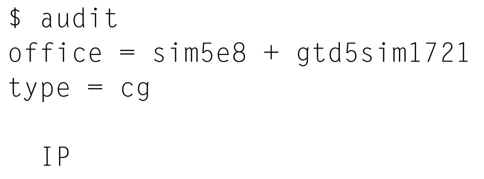
```
$ ctrlog olmb04 cg+rr active , all
```

In this example, the comma is used as a placeholder for the unused `trap` parameter. No placeholders are needed for the last two unused parameters (`start` and `stop`), because they are optional and the system will supply default values for them.

Prompting for parameters

When you enter a command at the *Linux* system shell, you may not be aware of the required parameters. Many commands, such as `audit`**,** will prompt for the parameters. If the parameter was not specified on the command line, then the name of the parameter requiring data will appear. At the prompt, the value or values for this parameter may be input. Figure 8 shows an example of prompting for the parameters for the `audit` command.

Figure 8 Prompting for parameters



```
$ audit
office = sim5e8 + gtd5sim1721
type = cg

  IP
```

Various error messages may be displayed during the prompting for parameters. Some of the more common messages are shown in Figure 9 through Figure 11.

Figure 9 Error message for invalid single office entry

```
$ audallow

office = badname

  ?E
INVALID OFFICE: badname does not exist.

Reenter value(s) for this keyword.
```

Figure 10 Error message for both valid and invalid office entries

```
$ audallow sim5e8+badname all
  ?E
INVALID OFFICE: badname does not exist.
Reenter remaining value(s) for keyword office
beginning with new value for 'badname'.

office =
```

In both of the preceding examples, the office ***badname*** does not exist. It may be misspelled or the office may have been removed from the database. In Figure 11, the office sim5e8 was accepted. The input for ***badname*** should be entered as ***goodname*+**sim5e8.

> **Important!** "`all`" need not be entered again. Only input for the office parameter is required.

Figure 11 Error message for invalid office type

```
office = $ audallow sim5e8 xxx
  ?E
INVALID TYPE: xxx is invalid; or this command is not valid for this
   office
type.
Reenter value(s) for this keyword.

type =
```

In Figure 11, the value "*xxx*" is either not valid or is not valid for this office type. Reenter the data for this parameter only.

> **Reference:** "Using parameters with control commands" (p. 5)

□

2 Commands for Auditing Network Elements

Overview

Purpose

This chapter lists and describes the NTM audit commands. Use audit commands to audit the switch and to inhibit or allow discrete-triggered audits.

> **Important!** See Table 2-1, “1A ESS TG valid values” (p. 47) for information required when coordinating with 1A *ESS* switch administrators.

Contents

This chapter contains the following topics:

□

Background

Overview

Purpose

This section defines audits and provides background information about how audits work.

Contents

This section contains the following topics:

□

What is an audit?

Description

An ***audit*** is an integrity check through which NTM searches both its own and the office's database for differences and corrects them if possible. The audits:

- synchronize the NTM database and the office database
- obtain results that may represent differences in the trunk groups or controls in the NTM and office databases

The results are generated into reports that can be routed to a file or to the screen (default).

NTM maintains a database in which all the offices of the supervised network are defined, as well as the data that will be collected from each office, and the status of any controls at the office. The offices each have their own records of data to be sent to NTM and of control status. These records must always be in agreement, so that NTM can receive and process the data accurately. There are two types of audits: regular and schedule.

Reference: "Audit types / matrix" (p. 30)

The network elements may be audited in a number of ways, depending on the source of the request and the current environment.

Audit error response information

See the "Audit error response information" section in the *System Responses Guide*.

□

Regular audits

Overview

A regular audit brings the appropriate data records to NTM from the office. NTM then compares the data in its current database with that sent from the office database. If differences are found, the office data updates the NTM data in the current database.

Regular audits are divided into two types:

- Manual audits — normal demand request to run audits
- Discrete-triggered audits — automatically requested, based on a stimulus from the switch

Important! Table 3 shows which audits are “manual”, “manual and discrete”, and “discrete” only.

Manual audits

You have the option of executing ***manually triggered*** audits whenever necessary. All manually triggered audits are executed from the *Linux* system shell by means of the `audit` command. Any differences found between databases are output to your screen or to a file. NTM supports three separate, simultaneous audit requests.

Audits need to be run after a full `create` in order to update the database fields for each of the network elements. Also, whenever you suspect that changes have been made in a network element while NTM was not receiving data, run manual audits for those associated network elements (e.g., after a single office `create`, after the system has been down [`stopsys`]).

Audits executed by switch type are usually performed more efficiently. Manual audits should be run on a switch-type basis such as: `audit all5e tglist`.

Specifying offices

To specify the offices for a manual audit, you can enter:

- Individual office names
- Office set names
- DCC office names
- A combination of any these

If you specify an office set name, all the offices in that set are audited. If you specify a DCC office name, all the offices connected to NTM through that DCC are audited. To audit a specific type of office, enter one of the values specified in the command syntax.

Important! If BDR is turned on, only the offices that are in primary or takeover mode can be audited.

All audit processes will be terminated when the stopsys command is executed.

Timing out

If any audit command times out before it successfully completes, it automatically tries three times, with successively longer timers, and aborts if the third try fails.

Important! This retry feature is not valid in NTM systems that support *GTD-5* switch types.

The audit all command is designed to provide capabilities to retry and log offices that fail or time out. The retry and logging functions are based on the reply received by the audit process and audit server. The "audit all" design is based on a two-pass approach.

- Pass 1 — The audit all process will run audits on all active offices based on the command parameters that the user enters (e.g., all, all5e, alldms, alldms250, allewsd). If an audit fails during pass 1, it will be placed on an internal list for use during pass 2.
- Pass 2 — The audit process will use the internal list of offices that failed and will run the audits for those offices. Audits that fail during this pass will be listed (along with the *clli*, audit type, and a comment) in the file:

"/musr/auds/aud_all.*time*" (where *time* is in the format HHMMSS)

audit all

The improved audit all command works for the following:

- audit all all
- audit all*office* all (all*office* can be all, all5e, alldms, alldms250, all4e, etc.)
- audit all all (*type* can be cg, tglist, cg+tg, etc.)
- combination of audit all*office* all

The format for the "/musr/auds/aud_all.*time*" is as follows:

```
audit vntrca1lds0 tg            # NO RESPONSE
audit vntrca1lds0 iec             # NOT RUN
audit vntrca1lds0 tgc             # NOT RUN
audit vntrca1lds0 cg              # NOT RUN
audit lsanca03ds2 pktsched      # TIMEOUT
audit lsanca03ds2 tgsched         # NOT RUN
audit lsanca03ds2 tg              # NOT RUN
audit lsanca03ds2 iec             # NOT RUN
audit lsanca03ds2 tgc             # NOT RUN
audit lsanca03ds2 cg              # NOT RUN
```

Important! With this file format, a user may make the file executable and run it as a shell program.

No response

A line in the file that ends with "# NO RESPONSE" indicates that a possible network or DCC problem was encountered while this audit was running. A line in the file that ends with a "# TIMEOUT" indicates that the audit timed out. A line in the file that ends with "# NOT RUN" indicates that this audit was not run because a previous audit failure occurred during Pass 2 on that office. For any of these situations, run `linkstat` to determine if the office is online, is active, and has good data, or if the DCC audit link has failed.

Files

It is the responsibility of the user to remove the "/musr/auds/aud_all.*time*" files. A `cron` job can be set up to remove the files after they are *n* days old, where *n* is the number of days.

In the "/musr/auds" directory, the ".status.*pid*" (where *pid* is the process ID) file will be created and updated whenever an `audit all` is run. During Pass 1, this file will contain one line that will indicate how many offices have been audited versus the total number of offices to be audited, and how many of the offices failed. For example:

```
17 of 127 entities attempted; 10 failed in Pass 1
```

During Pass 2, the first line will be the line from Pass 1 and the second line will contain the number of re-attempted offices out of the total number of previously failed offices, as well as how many of the re-attempted offices have failed for the second time. For example:

```
127 of 127 entities attempted; 17 failed in Pass 1
 10 of 17 entities retried; 10 failed in Pass 2
```

After Pass 2 has completed, the ".status.*pid*" file is moved to the ".status" file in the current directory. If the user wants to retain the information in the ".status" file, the file should be copied to another filename, because the ".status" file will be overwritten by the next `audit all`.

Running audits on a DCC

Manual audits usually should not be run on a DCC (Data Collection Concentrator) basis because this generally takes longer to complete. A manual audit may need to be performed, however, if a new DCC has been added or other changes have been made that affect all offices on a single DCC.

Manual audits run on a DCC basis have inherent delays. That is, each DCC can only process one audit at a time. Therefore, audits are queued to be run in sequence (one after another).

- For *GTD-5* offices, the `audit` command attempts the audit for 12 minutes before terminating the audit request if an office has not responded. The `audit` command will not retry the same audit after it is terminated.
- For non-*GTD-5* offices, if an office has not responded to an audit request in 5 minutes, then the `audit` command will retry the same audit for another 5 minutes before terminating the audit request. During the 10 minutes, audits can be queued to run — but no audits will be addressed by that DCC.

The `audit` command has the ability to send one audit to each DCC. Therefore, audits are run in parallel (single audits to multiple DCCs).

Discrete-triggered audits

Important! Discrete-triggered audits are not supported for the *GTD-5* switch type.

Discrete-triggered audits are requested by the network element. The switch, upon changing data that impacts the NTM system, sets a discrete to request an audit. If the discrete-triggered audit is allowed and discrete-triggered audits are supported, then NTM requests the audit. The switch clears the discrete after receiving the request for the audit. The `audstat`, `audinh`, and `audallow` commands are used to control discrete-triggered audits.

If the discrete is inhibited or not supported, then the discrete will remain set from the switch until the manual audit is run.

The Network Manager should view the "/musr/log/aud_disc" file periodically to see which discrete-triggered audit is running.

□

Schedule audits

Overview

A schedule audit works in the opposite direction from a regular audit. NTM sends updated schedule data to the office. The office compares the data in its database with the NTM data. If differences are found, the NTM data overwrites the office data in the office database as required.

A schedule audit is therefore a command that forces changes in an office database so that the information in it agrees with the information in the NTM database.

Using trunk group schedules

Schedules allow you to select the trunk groups for which you want to collect data from each office. There are limits by office type on the maximum number of trunk groups that can be scheduled. Also, some trunk groups identified at the office may not carry traffic; therefore, no data is available for them.

To schedule a trunk group for data collection, use the Trunk group schedule audit (tgsched) value for the `options` parameter in the Trunk Group File.

Reference: "Scheduling threshold tables" (p. 23) in the *Record Base Administration Guide*

When to run audits

You should run audits if:

- You have changed record base files.
- You have installed new databases.
- NTM has not polled the office for some time.
- You add or remove a network element.

Reference: Chapter 9, "Adding and Removing Network Elements" and Chapter 10, "Time Synchronization" in the *System Administration Guide*.

□

Updating the record base

Overview

Audits do not update the record base files. If these files need to be changed to reflect any differences reported by the audits, record base personnel must edit the *“/musr/rb”* files and then test, create, and install the changes.

> **Reference:** Chapter 8, “Record Base Update Procedures” in the *Record Base Administration Guide* and the `dbtest`, `create` and `installdb` commands.

Audit log files

Discrete-triggered audits

All audits triggered by discretes are logged in the *“/musr/log/aud_disc”* ASCII log file. The entry in the log file will name the office, the date and time, the name of the audit that was performed, and the reason for failure if an audit failed to complete.

Audits run by dayend command

All audits run by the `dayend` command are logged in the *“/musr/nmadm/audout.<date>”* ASCII log file. Each of these log files contains the complete audit output, as it would appear on the terminal screen. The <date> suffix is the day of the week on which the `dayend` command was run (Sun [Sunday] to Sat [Saturday]), so a maximum of 7 logs will be stored in this directory.

Audits run manually

Audit output for manual audits is directed to the user's screen, but it can be redirected to a save file. A manually-initiated audit will time-out in 5 minutes. If more than 5 minutes elapse before the audit completes, the prompt returns to the screen and the output from the audit appears only in the *“/musr/log/aud_user”* file.

Audits run by recreate command

The `recreate` command tests, creates, and installs touched files in the record base. It is normally set to run as a part of the `dayend` routines, but it can also be run as a manual command. Output from audits initiated by `recreate` are output to the *“/musr/log/recreate.out”* log file.

□

Audit commands

Overview

Purpose

This section provides detailed information about the audit commands.

Table

Table 1 lists each NTM audit command and its function.

Table 1 Audit commands and functions

Command	Function
`audit`	Audits a switch or set of switches for any or all audit types
`audallow`	Enables discrete-triggered audits
`audinh`	Disables discrete-triggered audits
`audstat`	Obtains status of discrete-triggered audits
`local_audit`	Provides a quick approximation of an `audit all` on a switch
`udaudit`	Audits a UDNE switch or set of switches for any or all audit types

Contents

This section contains the following commands:

□

audallow

Description

The audallow (audit allow) command enables discrete-triggered audits from the switch.

Syntax

```
audallow office=office type=types
```

Switch restrictions

Discrete triggered audits are not supported for the *GTD-5* switch.

Parameters

You can enter multiple values for each of these parameters.

`office` One of the following:

- Office name(s)
- all1a — all 1A *ESS* switch offices
- all4e — all *4ESS* switch offices
- all5e — all *5ESS* switch offices
- alldms — all *DMS* 100/200 switch offices
- alldms250 — all *DMS* 250 switch offices
- alldms300 — all *DMS* 300 switch offices
- alldms500 — all *DMS* 500 switch offices
- allscsnsn — all Succession offices
- alldcc — all DCC [Data Collection Concentrator] switch offices
- allewsd — all *EWSD* switch offices
- allgsp — all GSP offices
- allgsx — all Sonus GSX offices
- allpsx — all Sonus PSX offices
- all_lssgr — all LSSGR offices
- all

When more than one office is specified (for example, all4e) and one or more of the CLLIs are valid, activated, and successful, then the command will return successfully. A failure is reported only if all CLLIs are invalid, are deactivated, or have failed.

`types` Audit type(s). See Table 3, “types” (p. 30) for audits having a trigger type of A or D.

System responses

When you enter the `audallow` command, the system returns the following response.

```
Audit status updated.
```

Examples

```
$  audallow  office=clev type=cg
```

□

audinh

Description

The `audinh` (audit inhibit) command disables discrete-triggered audits from the switch.

Syntax

```
audinh office=office type=types
```

Switch restrictions

1. Discrete triggered audits are not supported for the *GTD-5* switch.
2. NTM does not support Phase 1 *DMS* 100 switches. If you connect a Phase 1 *DMS* 100 switch to NTM, you should use the `audinh` command to inhibit discrete-triggered trunk group reference data audits from running. You should also set up a shell script to manually run trunk group reference data audits on a daily basis.

Parameters

You can enter multiple values for each of these parameters.

`office` One of the following:

- Office name(s)
- all1a — all 1A *ESS* switch offices
- all4e — all *4ESS* switch offices
- all5e — all *5ESS* switch offices
- alldms — all *DMS* 100/200 switch offices
- alldms250 — all *DMS* 250 switch offices
- alldms300 — all *DMS* 300 switch offices
- alldms500 — all *DMS* 500 switch offices
- allscsnsn — all Succession offices
- alldcc — all DCC [Data Collection Concentrator] switch offices
- allewsd — all *EWSD* switch offices
- allgsx — all Sonus GSX offices
- allpsx — all Sonus PSX offices
- all_lssgr — all LSSGR offices
- all

When more than one office is specified (for example, all4e) and one or more of the CLLIs are valid, activated, and successful, then the command will return successfully. A failure is reported only if all CLLIs are invalid, are deactivated, or have failed.

`types`	Audit type(s). See Table 3, “types” (p. 30) for audits having a trigger type of A or D.

System responses

When you enter the `audinh` command, the system returns the following response.

```
Audit status updated.
```

Examples

```
$  audinh  office=abcd type=cg
```

□

audit

Description

The `audit` command audits the switch for all audit types. See Table , "Audit types / matrix" (p. 30) to determine which audit types can be run manually and which audits are triggered by a discrete. To specify the offices for a manual audit, you may specify:

- Individual office names
- Office set names — all the offices in that set will be audited.
- DCC (Data Collection Concentrator) office names — all the offices connected to NTM through that DCC are audited.

  Manual Audits run on a DCC basis have inherent delays. That is, each DCC can only process one audit at a time. Therefore, audits are queued to be run in sequence (one after another).

  – For *GTD-5* offices, the audit command attempts the audit for 12 minutes before terminating the audit request if an office has not responded. The audit command will not retry the same audit after it is terminated.

  – For non-*GTD-5* offices, if an office has not responded to an audit request in 5 minutes, then the audit command will retry the same audit for another 5 minutes before terminating the audit request. During the 10 minutes, audits can be queued to run — but no audits will be addressed by that DCC.
- A combination of any of the above

Syntax

```
audit office=office type=types
```

Feature restrictions

If Feature 8, "Disaster Recovery (Duplex)" and Feature 40, "Enhanced Disaster Recovery" is turned on, only the network elements that are in Primary or takeover mode may be audited.

Restrictions

A maximum of 3 audits can be run simultaneously.

Parameters

NTM does not support Phase 1 *DMS* 100 switches. If you intend to connect a Phase 1 *DMS 100* switch to NTM, we recommend that you use the `audinh` command to inhibit discrete-triggered trunk group reference data audits from running. You should also set up a shell script to manually run trunk group reference data audits on a daily basis.

You can enter multiple values for each of these parameters.

`office` One or more of the following:

- Office name(s)
- all1a — all 1A *ESS* switch offices
- all4e — all *4ESS* switch offices
- all5e — all *5ESS* switch offices
- alldms — all *DMS* 100/200 switch offices
- alldms250 — all *DMS* 250 switch offices
- alldms300 — all *DMS* 300 switch offices
- alldms500 — all *DMS* 500 switch offices
- alldcc — all DCC switch offices
- allewsd — all *EWSD* switch offices
- all_lssgr — all LSSGR switch offices
- allgtd5 — all *GTD-5* switch offices
- allgsx — all Sonus GSX switch offices
- allpsx — all Sonus PSX switch offices
- all

`types` Audit type(s). See Table 3, “types” (p. 30).

System responses

When you enter the audit command, the system returns various types of data depending upon which audit type is requested and which network elements as queried. Information the audit finds that was added to the record base is reflected in the screen output.

> **Reference:** See “Error messages” (p. 18) for errors that can be displayed while running this command.

Cancelling queued audits

To cancel an audit that has been queued to a switch:

- use `deact` to drop the link to the switch
- use `act` to reactivate the link to the switch

Dropping the link will cancel ***ALL*** of the requested audits for that switch.

Examples

```
$  audit  office=all5e type=silc
```

Error messages

The following error message is generated when the entity input as the office cannot have audits allowed at this time.

```
?E MISSING DATA: keyword office needs data; or the office is
   deactivated.
```

This message may appear for the following reasons:

- The office is an invalid entry.
- The office is deactivated.
- The office link has failed.
- There are no offices of this type active at this time (for example, if you specified `all1a`, and no 1A *ESS* offices are active).

Use `linkstat` to determine the status of the desired office.

References

See “recreate” (p. 63) for information on audits that are run during the `dayend` procedure.

See Table 2-1, “1A ESS TG valid values” (p. 47) for information required when coordinating with 1A *ESS* switch administrators.

□

audstat

Description

The `audstat` (audit status) command displays the discrete-triggered audits currently allowed or inhibited on the switch.

Syntax

```
audstat  office=office
```

Switch restrictions

Discrete triggered audits are not supported for the *GTD-5* switch.

Parameters

`office` One or more of the following:

- office name
- all1a — all 1A *ESS* switch offices
- all4e — all *4ESS* switch offices
- all5e — all *5ESS* switch offices
- alldms — all *DMS* 100/200 switch offices
- alldms250 — all *DMS* 250 switch offices
- alldms300 — all *DMS* 300 switch offices
- alldms500 — all *DMS* 500 switch offices
- allscsnsn — all Succession offices
- alldcc — all DCC [Data Collection Concentrator] switch offices
- allewsd — all *EWSD* switch offices
- allgsx — all Sonus GSX offices
- allpsx — all Sonus PSX offices
- all_lssgr — all LSSGR offices
- all

Important! You can enter multiple values separated by the plus sign (+).

System responses

Figure 1 shows the output for the `audstat` command. In this figure, “alw” indicates that the discrete-triggered audit is allowed. If a particular audit is inhibited, “inh” will appear in that column instead.

Figure 1 audstat command output

```
IP

               1A ESS DISCRETE AUDIT STATUS
               ---------------------------
CLLI           REF  PP   TGC  CG   SILC

troyrs02a1h    alw  alw  alw  alw  alw
troyru46a1h    alw  alw  alw  alw  alw
troygr46a1h    alw  alw  alw  alw  alw
sim1ae10       alw  alw  alw  alw  alw


               4ESS DISCRETE AUDIT STATUS
               --------------------------
CLLI           TGINFO  CG   RR   HTR

sim4e13ent01    alw    alw  alw  alw
sim4e12ent06    alw    alw  alw  alw
sim4e13ent05    alw    alw  alw  alw
sim4e12ent02    alw    alw  alw  alw
sim4e14ent03    alw    alw  alw  alw

                              5ESS DISCRETE AUDIT STATUS
               --------------------------------------------------------
CLLI           PKTSCHED  TGSCHED   TG     TGC  CG   ATG  SILC  IEC  RSM

npvlda06e5h      alw       alw     alw    alw  alw  alw  alw   alw  alw
dba5e5ent12      alw       alw     alw    alw  alw  alw  alw   alw  alw
dba5e6ent13      alw       alw     alw    alw  alw  alw  alw   alw  alw
dba5e5ent11      alw       alw     alw    alw  alw  alw  alw   alw  alw
npvlda28e5h      alw       alw     alw    alw  alw  alw  alw   alw  alw

DMS 100 DISCRETE AUDIT STATUS
               ----------------------------------------
CLLI           PKTSCHED  TGSCHED   TG     TGC  CG   IEC

clmbrs12dmc      alw       alw     alw    alw  alw  alw
clmbda12dmc      alw       alw     alw    alw  alw  alw
clmbda13dmc      alw       alw     alw    alw  alw  alw
simdms24         alw       alw     alw    alw  alw  alw


                    DMS-250 DISCRETE AUDIT STATUS
              -------------------------------------
CLLI          PKTSCHED  TGSCHED  TG   TGC  CG   IEC

dms250_250f     alw       alw    alw  alw  alw  alw
```

```
dms250_250t     alw      alw    alw  alw  alw  alw
dms250_250d     alw      alw    alw  alw  alw  alw
dms250_1024f    alw      alw    alw  alw  alw  alw
dms250_1024t    alw      alw    alw  alw  alw  alw

                    EWSD DISCRETE AUDIT STATUS
               ----------------------------------------
CLLI           TGSCHED   TG     TGC  CG

ewsd10sim        alw     alw    alw  alw


                    LSSGR DISCRETE AUDIT STATUS
               ----------------------------------------
CLLI           PKTSCHED  TGSCHED   TG     TGC  CG   IEC

lssgrsim         alw       alw     alw    alw  alw  alw
```

□

local_audit

Description

The local_audit command provides a quick approximation of “audit all all” by using the results of the last audit command for each audit type. The benefit of the local_audit command is that it runs much faster than the audit command and gets the database back in sync.

> **Important!** New trunk groups added to the record base file do not appear in 5-minute data until a regular audit is done. Therefore, the individual use of local_audit should always be followed by the equivalent audit command.

Syntax

```
local_audit all office
```

Feature restrictions

This command is available only with the purchase of Feature 86, “Local Audit Data Restoration”.

Parameters

all	all offices (this is the default)
office	CLLI name — name of office to be audited

System responses

When you enter the local_audit command, the system returns various types of data depending upon which audit type is requested and which network elements as queried.

Examples

```
$  local_audit  okld
```

References

See the “RSPTE File” (p. 68) in the *Record Base Administration Guide* for more information on defining the hostname.

□

udaudit

Description

The `udaudit` command audits the UDNE switches for all audit types. See Table , "Audit types / matrix" (p. 30) to determine which audit types can be run.

> **Important!** To execute all the audits for all switches in the system execute `audit all all` and then `udaudit all all`.

Syntax

```
udaudit office=office type=types
```

Feature restrictions

This command is only available if you have features:

Feature 436, "UDDM/UDNEI",

Feature 438, "Support for NexTone Session Border Controller",

Feature 455, "Support for NexTone Session Border Controller Outbound Call Limiting".

Restrictions

A maximum of 3 udaudits can be run simultaneously.

Parameters

You can enter multiple values for each of these parameters.

office	One or more of the following:

- Office name(s)
- allnextone — all *nextone* switch offices
- all

types	Audit type(s). See Table 3, "types" (p. 30).

System responses

When you enter the `udaudit` command, the system returns various types of data depending upon which audit type is requested and which network elements as queried. Information the audit finds that was added to the record base is reflected in the screen output.

UDAUDIT Configuration File

The following are the udaudit tracing and configuration parameters from the */nm/etc/udaudit.config* file.

Important! Editing this file is not allowed without the assistance from Alcatel-Lucent field support.

- udaudit.logSize - Size of the log file. The default is 1,000,000.
- udaudit.logRotate - Number of the log files. The default is 2.
- udaudit.logFormat - Format of the log files: TEXT or XML. The default is TEXT.
- udaudit.logLevel - The logging level. It is updated every 5 minutes. One of the following option: SEVERE, WARNING, INFO, CONFIG, ALL, OFF. The default is OFF.
- udaudit.logConsoleLevel - The console logging level. The default is OFF.

UDAUDIT Server Configuration File

The following are the udaudit server tracing and configuration parameters from the */nm/etc/UDAUDSRV.config* file.

Important! Editing this file is not allowed without the assistance from Alcatel-Lucent field support.

- UDAUDSRV.logSize - Size of the log file. The default is 1,000,000.
- UDAUDSRV.logRotate - Number of the log files. The default is 2.
- UDAUDSRV.logFormat - Format of the log files: TEXT or XML. The default is TEXT.
- UDAUDSRV.logLevel - The logging level. It is updated every 5 minutes. One of the following option: SEVERE, WARNING, INFO, CONFIG, ALL, OFF. The default is OFF.
- UDAUDSRV.logConsoleLevel - The console logging level. The default is OFF.
- UDAUDSRV.timeOut - Timeout value in milliseconds. The default is 30000.

Examples

```
$  udaudit  office=nextone_1 type=sbclist
```

Error messages

The following error message is generated when the entity input as the office cannot have audits allowed at this time.

```
INPUT ERROR: [nextone_1] is invalid 'office'.
Reenter value(s) for this keyword.
```

This message may appear for the following reasons:

- The office is an invalid entry.
- The office is deactivated.
- The office link has failed.
- There are no offices of this type active at this time.

Use `linkstat` to determine the status of the desired office.

The following error:

```
udaudit: Failed to send request to UDAUDSRV.
```

occurs when udaudit command could not send message to UDAUDSRV. The serve did not acknowledge the message. To solve the problem verify whether UDAUDSRV is up and running.

The following error:

```
udaudit: Timeout on audit response from: 'pnnxt_1'
```

occurs when udaudit did not get resposne from UDAUDSRV in a time specified in */nm/etc/UDAUDSRV.config* file. Verify whether UDAUDSRV has enough resources to execute its tasks on time.

The following error:

```
UDAUDSRV: switch 'pnnxt_1', audit 'sbclist':UDACSRV: Timeout.
   Did not get response from DCOL server.
```

occurs when udaudit did not get resposne from UDNE DCOL in a time specified in */nm/etc/UDAUDSRV.config* file. Increase timeout in the configuration file or refer to UDNE DCOL documentation.

The following response:

```
UDAUDSRV: switch 'pnnxt_1', audit 'sbclist':
DCOL: SOAP: Config does not match schema.
```

appears when error occurred in UDNE DCOL.

References

See “recreate” (p. 63) for information on audits that are run during the `dayend` procedure.

□

Audit descriptions and examples

Overview

Purpose

This section provides descriptive information and example output for each of the different audit types.

Audit output indicates any errors that occurred during the audit and differences between the office and NTM databases that should be manually resolved. If there are no problems, most of the audits will return you to the `$` prompt, indicating satisfactory completion of the command.

Table

Table 2 shows the `audit` system response.

Table 2 audit system response

IF the command ...	THEN the system responds with ...
executes successfully	"IP" followed by the output for the audit type(s) specified
does not execute successfully	the appropriate error message

Contents

This section contains the following topics:

□

Audit types / matrix

Restrictions

NTM does not support Phase 1 *DMS* 100 switches. If you intend to connect a Phase 1 *DMS* 100 switch to NTM, we recommend that you use the `audinh` command (2-14) to inhibit discrete-triggered trunk group reference data audits from running. You should also set up a shell script to manually run Trunk group reference audit (tg) audits on a daily basis.

Matrix

Table 3 provides a list of the audit types, their expanded names, and the office types that support them. The following key applies to this table.

Key

- M = Audit available for user (manual)
- A = Audit available for user (manual) and automatically audited by discrete (discrete-triggered)
- D = Discrete-triggered only (no manual available)

Table 3 types (Sheet 1 of 2)

			see Key															
							DMS			SCSN								
Type	**Expanded Name**	**See page**	**1A *ESS***	**4ESS**	***4ESS* Block #**	**5ESS**	**NA16 and later**	**UCS16 and later**	**sn04tdm**	**sn02 and later**			**GTD-5**	**Sonus GSX**	**Sonus PSX**	**GSP**	**EWSD**	**UDNE***
atg	Automatic Trunk Group Control	(2-32)		See tginfo		A								A				
cg	Call Gap Code	(2-34)	A	A	46	A	A	A	A	A	A	A	M		A			
cni	Common Network Interface Configuration Data	(2-36)		M 4e14+	55													
dpt	Dynamic Packet Trunks	(2-38)								A sn04+								
eplist	SignalingEndpnt reference data list	(2-40)																M
htr	Hard-to-Reach Controls	(2-41)		A	10, 12–16, 20–27	A 5e16_1h+	A	A	A	A sn03+			M					
iec	Interexchange Carrier Trunk Group Reference	(2-43)				M		M	M	M	M	M 13+						
mtd	Modified Trunk Subgroup Data	(2-48)		via tgsched														
ocldtn	Dialed Telephone Number	(2-51)																M
oclendpnt	End Point Web Service	(2-52)																M
oclrealm	Realm Web Service	(2-53)																M

Table 3 types (Sheet 2 of 2)

			see Key															
							DMS			SCSN								
Type	**Expanded Name**	**See page**	**1A *ESS***	**4ESS**	***4ESS* Block #**	**5ESS**	**NA16 and later**	**UCS16 and later**	**sn04tdm**	**sn02 and later**			**GTD-5**	**Sonus GSX**	**Sonus PSX**	**GSP**	**EWSD**	**UDNE***
pktsched	Packet Schedule	(2-54)				A	A	A	A	A	A	A 13+		A	M			
pp	Preprogram Trunk Group Control	(2-56)	A															
pplist	Preprogram Trunk Group Control List	(2-57)	M															
ratelimit	Ratelimit audit type	(2-59)																M
ref	Machine and Trunk Group Reference **NOTE:** See Table 2-1, “1A ESS TG valid values” (p. 47) for information required when coordinating with 1A *ESS* switch administrators.	(2-45)	A															
rr	Reroute	(2-62)		A	49-51													
rlmlist	SignalingRealm reference data audits list	(2-72)																M
sbclist	SBC reference data audits list	(2-64)																M
rsm	Remote Switching Module	(2-61)				A												
silc	Selective Incoming Load Controls	(2-66)	A			A							via tgc					
sls	Signaling Linkset	(2-68)				A 5e16_1h+												
tg	Trunk Group Reference	(2-86)				A	A	A	A	A	A	A	M	A		A		
tgc	Trunk Group Control **NOTE:** For *GTD-5* offices, this audit is equivalent to the tgc, atg, and silc audits.	(2-74)	A	See tginfo		A	A	A	A	A	A	A	M	A		A		
tginfo	Trunk Group Reference and Control **NOTE**: For *4ESS*, this performs the tgc and atg audits.	(2-88)		M	35													
tglist	Trunk Group List	(2-77)	M	M 4e18+	47	M5e8+	M	M	M	M	M	M	M	M				
tgsched	Trunk Group Schedule **NOTE:** For *GTD-5* offices, this audit internally requests the **tgschedlist** audit before and after the tgsched audit.	(2-90)		A 4e17+	24	A	A	A	A	A	A	A	M	A				
tgschedlist	Trunk Group Schedule List	(2-93)																
wsdl	wsdl audit	(2-96)																M
all	All office audit types in succession (except tglist, iec)																	

* The audits on the UDNE switches can be only executed with `udaudit` command.

Automatic trunk group control audit (atg)

Description

The automatic trunk group control audit requests data from the *5ESS*, 7R/E, and *GTD-5* switches on active automatic trunk group controls. You can request this audit manually or it will be triggered automatically by a discrete for the *5ESS* and 7R/E switch.

For *4ESS*, those audits are done in Trunk group reference and control audit (tginfo). For the *GTD-5* switch, this audit is run as part of the Trunk group control audit (tgc). Any differences found by the audit are used to update the NTM database. The automatic trunk group controls are:

- ACC
- CR (5e6 and later; ivtoa3 and later)

You can enable these controls at any time; however, the control will not be activated until a specified threshold is exceeded.

Sample output

Figure 2 is an example of output for the automatic trunk group control audit.

Figure 2 Automatic trunk group control audit output

```
$ audit dba5e6ent13 atg
  IP
Thu Jun 24 15:09:24 1993
dba5e6ent13 : 5ESS AUTOMATIC TRUNK GROUP CONTROL AUDIT
acc    from=dba5e6ent13 tg=sim4e14ent03-001 act=del
     ih/rs=rs sk/cn=cn cat=a
cr     from=dba5e6ent13 tg=sim4e14ent03-015 act=del
     ih/rs=rs sk/cn=cn cat=b ara=no l1=5 l2=10
acc    from=dba5e6ent13 tg=sim4e14ent03-002 act=add
     ih/rs=rs sk/cn=sk cat=d
cr     from=dba5e6ent13 tg=sim4e14ent03-012 act=add
     ih/rs=rs sk/cn=sk cat=a ara=no l1=5 l2=12
cr     from=dba5e6ent13 tg=sim4e14ent03-019 act=add
     ih/rs=rs sk/cn=cn cat=c ara=no l1=10 l2=15
$
```

This sample shows:

- automatic trunk group controls removed from the switch that are still present in the NTM database
- existing automatic trunk group controls in the switch that are not reflected in the NTM database

Important! Those trunk group controls deleted from the NTM database (`act=del`) are always listed first.

Possible errors

The following error message will be printed out by the `audit` command if a tgid was not found in the NTM database:

```
<switch> tgid : <tgid> - trunk group not found in database
```

This error could be caused by an incorrect keyword, incorrect spelling of a keyword, or a similar situation. Corrective actions include making these changes to the record base, doing a `create` and `installdb` of the database, and rerunning the audit.

□

Call gap control audit (cg)

Description

The call gap control audit checks for discrepancies in the call gap controls. Any differences are used to update the NTM database. You can request this audit manually or it will be triggered automatically by a discrete.

The sample output in Figure 3 and Figure 4 shows both:

- Call gap controls that show up in the NTM database and no longer exist in the switch. Deletions always appear first (`act=del`).
- Call gap controls that have been added to the switch but are not reflected in the NTM database

If no controls are listed, the NTM database is already current for that network element. This does not necessarily mean that no controls are active for that network element.

Sample output

Figure 3 is an example of the output for the call gap control audit for the *GTD-5* switch.

Figure 4 is an example of the output for the call gap control audit for other switch types.

Figure 3 Call gap control audit output for the GTD-5 switch

```
$ audit jantestent1 cg
  IP
Fri Feb 18 07:57:14 1994
jantestent1 : GTD-5 CALL GAP AUDIT
cg    office=jantestent1 act=add cg_typ=dest code=302420
        gap=5.4 pfxt=any annc=roa
cg    office=jantestent1 act=add cg_typ=dest ic=0113
        code=304854 gap=5.4 pfxt=any annc=fea
cg    office=jantestent1 act=add cg_typ=dest ic=5774
        gap=5.4 annc=fea
cg    office=jantestent1 act=add ic=5774 rc=172
        gap=5.4 annc=fea
cg    office=jantestent1 act=add ic=0113 rc=182
        ccd=352 gap=5.4 annc=rot
cg    office=jantestent1 act=add ic=0113 isrc=09
        gap=5.4 annc=fea
cg    office=jantestent1 act=add ic=0113 isrc=02
        ccd=49 gap=5.4 annc=rot
cg    office=jantestent1 act=add cg_typ=term code=3098548061
        lgap=10 annc=roa
$
```

Figure 4 Call gap control audit output for multiple switch types

```
$ audit sim1ae10+sim4e13ent01+dba5e6ent13+simdms24 cg
  IP
Thu Jun 24 15:09:24 1993
sim1ae10 : 1AESS CALL GAP AUDIT
cg    office=sim1ae10 act=del code=614
        gap=600 annc=nca
cg    office=sim1ae10 act=del ic=123 code=6145551
        gap=all annc=nca
cg    office=sim1ae10 act=add code=6142687
        gap=all annc=ea2
cg    office=sim1ae10 act=add code=61475545
        gap=all annc=ea1
cg    office=sim1ae10 act=add code=614821277
        gap=1200 annc=nca
cg    office=sim1ae10 act=add ic=399 code=614
        gap=300 annc=ea1
sim4e13ent01 : 4ESS CALL GAP AUDIT
cg    office=sim4e13ent01 act=del code=6145552004
     gap=50 annc=ea2 type=dom trap=off tg=none
cg    office=sim4e13ent01 act=add code=6145552001
     gap=all annc=nca type=dom trap=off tg=none
cg    office=sim4e13ent01 act=add code=6145552005
     gap=1200 annc=ssb type=dom trap=off tg=none
dba5e6ent13 : 5ESS CALL GAP AUDIT
cg    office=dba5e6ent13 act=del ic=123 code=61455521
        gap=1200 annc=ea2 dom=all
cg    office=dba5e6ent13 act=del code=6145552005
        gap=600 annc=ea1 dom=all
cg    office=dba5e6ent13 act=add code=61455525
        gap=300 dom=all annc=ea2
cg    office=dba5e6ent13 act=add ic=321 code=6145553
        gap=1200 annc=ea1 dom=all
simdms24 : DMS 100 CALL GAP AUDIT
cg    office=simdms24 act=del ic=123 code=6145552002
        gap=MIS annc=ea2
cg    office=simdms24 act=del ic=123 code=61455521
        gap=1200 annc=ea2
cg    office=simdms24 act=del code=6145552005
        gap=600 annc=ea1
cg    office=simdms24 act=del ic=123 code=6145552012
        gap=MIS annc=nca
cg    office= act=none ic=123 code=6145xxx
        gap=1200 annc=ea2 dom=cblb  : invalid response
cg    office= act=none ic=yyy code=61455521
        gap=1200 annc=ea2 dom=cblb  : invalid response
$
```

□

CNI configuration data audit (cni)

Description

The CNI configuration data audit provides descriptive data on the structure, capabilities, and interfaces for signaling messages. This data includes a list of nodes (equipped and unequipped) on the CNI and their respective node types.

> **Important!** The CNI configuration data audit is provided for 4e14 and subsequent generics only.

Sample output

Figure 5 is an example of a CNI Configuration Audit Output.

Figure 5 CNI configuration data audit output

```
$ audit sim4e14ent03 cni
  IP
Thu Jun 24 15:09:24 1993
sim4e14ent03 : 4ESS CNI CONFIGURATION AUDIT
cni   node id=1 type=ccitt7 act=add
      equip=no oos=yes
      office=artnvasm11t
cni   node id=2 type=ccitt7 act=add
      equip=yes oos=yes
      office=artnvasm12t
cni   node id=3 type=ccitt7 act=add
      equip=yes oos=no
      office=artnvasm03t
cni   node id=4 type=ccitt7 act=add
      equip=yes oos=no
      office=artnvasm04t
cni   node id=5 type=ccitt7 act=add
      equip=yes oos=no
      office=artnvasm05t
cni   node id=6 type=ccs7 act=add
      equip=yes oos=yes
      office=artnvasm06t
cni   node id=7 type=ccs7 act=add
      equip=yes oos=no
      office=artnvasm07t
cni   node id=8 type=ccitt7 act=add
      equip=yes oos=no
      office=artnvasm08t
cni   node id=9 type=ccs7 act=add
      equip=yes oos=no
```

```
          office=artnvasm09t
$
```

□

Dynamic packet trunks audit (dpt)

Description

The DPT (Dynamic Packet Trunks) audit checks for discrepancies in the DPT information and updates the NTM database. The output of the DPT audit consists of the following information:

- Max Tids
- Reservation
- Prioritization controls

You can request this audit manually or it will be triggered automatically by several discretes.

Feature restrictions

This audit type is only available with the purchase of Feature 365, “Bandwidth Directionalization & Prioritization control support in Succession Network Switch Generic SN04”.

Sample output

Figure 6 is an example of the output for the dpt audit.

Figure 6 DPT audit output

```
$ audit scsnsn04

type = dpt
 IP
Fri Dec 14 15:50:00 2001
scsnsn04 : SCSNSN DPT MAX TID AUDIT
scsnsn04 ctltype=dpttid action=add dptmaxtid=65537
Fri Dec 14 15:50:05 2001
scsnsn04 : SCSNSN DPT RESERVATION AUDIT
scsnsn04 ctltype=dptrs action=add dpt_thresh=74
Fri Dec 14 15:50:11 2001
scsnsn04 : SCSNSN DPT PRIORITY AUDIT
dptpri  from=scsnsn04 tg=testohdmsc-010 act=add
              spthr=74 remthr=74
dptpri  from=scsnsn04 tg=testoh1a8b-011 act=add
              spthr=75 remthr=75
dptpri  from=scsnsn04 tg=testoh1a9b-012 act=add
              spthr=76 remthr=76
dptpri  from=scsnsn04 tg=testohdmsc-020 act=add
              spthr=77 remthr=77
```

```
$
```

□

SignalingEndpnt audit (eplist)

Description

The NTM allows users to execute manual eplist audit type on Network Elemens of switchtype “nextone”. You can execute an eplist audit on individual nextone elements, nextone elements in a network element set, or all nextone elements.

Feature restrictions

This audit type is only available with the purchase of Feature 438, “Support for NexTone Session Border Controller”.

Sample output

Figure 7 is an example of the output for the eplist audit.

Figure 7 eplist audit output

```
$ audit scsnsn04
```

□

Hard-to-reach audit (htr)

Description

The HTR (Hard-To-Reach) audit checks for discrepancies in the HTR information and updates the NTM database. The HTR audit information consists of thresholds, assignments, and controls. You can request this audit manually or it will be triggered automatically by a discrete.

Sample output

Figure 6 is an example of HTR Domestic Audit Output for the *4ESS* switch.

Figure 8 HTR audit output

```
$ audit anhmca0295t htr
  IP
Mon Oct 31 08:41:07 1994
anhmca0295t : 4ESS INTERNATIONAL HTR DATA (MANUAL)
Mon Oct 31 08:41:14 1994
anhmca0295t : 4ESS DOMESTIC HTR DATA (MANUAL)
Mon Oct 31 08:41:29 1994
anhmca0295t : 4ESS DOMESTIC HTR DATA (REMOTE & MANUAL)
Mon Oct 31 08:41:35 1994
anhmca0295t : 4ESS DOMESTIC HTR NXX RES ASSIGNMENTS
Mon Oct 31 08:41:42 1994
anhmca0295t : 4ESS INTERNATIONAL COUNTRY CODES - THRESH SETS
Mon Oct 31 08:41:48 1994
anhmca0295t : 4ESS INTERNATIONAL THRESH SET ASSIGNMENTS
Mon Oct 31 08:41:54 1994
anhmca0295t : 4ESS NPA FAILURE TYPE ASSIGNMENTS
Mon Oct 31 08:42:00 1994
anhmca0295t : 4ESS HNPA FAILURE TYPE ASSIGNMENTS
Mon Oct 31 08:42:09 1994
anhmca0295t : 4ESS FIRST FNPA FAILURE TYPE ASSIGNMENTS
Mon Oct 31 08:42:16 1994
anhmca0295t : 4ESS SECOND FNPA FAILURE TYPE ASSIGNMENTS
Mon Oct 31 08:42:22 1994
anhmca0295t : 4ESS THIRD FNPA FAILURE TYPE ASSIGNMENTS
Mon Oct 31 08:42:29 1994
anhmca0295t : 4ESS FOURTH FNPA FAILURE TYPE ASSIGNMENTS
Mon Oct 31 08:42:35 1994
anhmca0295t : 4ESS FIFTH FNPA FAILURE TYPE ASSIGNMENTS
Mon Oct 31 08:42:41 1994
anhmca0295t : 4ESS SIXTH FNPA FAILURE TYPE ASSIGNMENTS
$
```

Figure 9 is an example of HTR Domestic Audit Output for the *GTD-5* switch.

Figure 9 HTR audit output for the *GTD-5* switch

```
$ audit eriepa007gt htr
  IP
Thu May 12 09:04:36 1994
eriepa007gt : GTD-5 MANUAL HARD-TO-REACH CONTROL AUDIT
dhtr  office=eriepa007gt act=add ic=1234 npa=614
        nxx=860 list=clm ftype=b
dhtr  office=eriepa007gt act=add ic=3334 npa=708
        nxx=257 list=clm ftype=b
$
```

Possible errors

The following error message will be printed by the `audit` command if a tgid was not found in the NTM database:

<*switch*> tgid : <*tgid*> - trunk group not found in database

This error could be caused by an incorrect keyword, incorrect spelling of a keyword, etc. Corrective actions include making these changes to the record base, doing a `create` and `installdb` of the database, and rerunning the audit.

□

Interexchange (InterLATA) carrier audit (iec)

Description

The interexchange (or interLATA) carrier audit requests information related to trunk groups used to reach interLATA carrier switches and updates the NTM database with this information. This audit can be requested manually or it may be triggered by a discrete. The output consists of carrier prefixes that have been added to or deleted from the NTM database. The data received from this audit refers only to the interexchange carrier packet of the 5-minute response.

Sample output

Figure 10 is an example of output for the `iec` audit.

Figure 10 Interexchange carrier trunk group reference audit output

```
$ audit dba5e6ent13+simdms24 iec
  IP
Thu Jun 24 15:09:24 1993
dba5e6ent13 : 5ESS INTEREXCHANGE CARRIER TRUNK GROUP REFERENCE AUDIT
        carrier prefix 93 added to active database schedule
        carrier prefix 94 added to active database schedule
        carrier prefix 98 added to active database schedule
        carrier prefix 89 (feature group D) not found in database
        carrier prefix 92 deleted from active database schedule
simdms24 : DMS 100 INTEREXCHANGE CARRIER TRUNK GROUP REFERENCE AUDIT
        carrier prefix 55 added to active database schedule
        carrier prefix 76 added to active database schedule
        carrier prefix 78 added to active database schedule
        carrier prefix 51 deleted from active database schedule
        carrier prefix 58 deleted from active database schedule
$
```

This sample shows any changes in the scheduling of IEC (Interexchange Carrier) trunk groups for 5-minute data. Those IEC trunk groups recently scheduled are listed first. These trunk groups are followed by IEC trunk groups that have been removed from the schedule.

Possible errors

The error message, “carrier prefix 89 (feature group D) not found in database,” indicates one of the following:

- the carrier prefix is not defined in the NTM database

- the carrier prefix is defined in the NTM database but the feature group has not been defined correctly

□

Machine and trunk group reference audit (ref)

Description

This audit is only used with the 1A *ESS* switches to update the NTM database with reference information for the 5-minute data message machine and trunk group counts.

> **Important!** Table 2-1, “1A ESS TG valid values” (p. 47) provides information required when coordinating with 1A *ESS* switch administrators.

If the audit runs successfully, no output other than the audit title and a list of trunk groups added to or deleted from the database schedule are provided.

> **Important!** The audit must be run on any 1A *ESS* switches before 5-minute data collection will take place successfully.

Sample output

Figure 11 illustrates the output for the Machine and Trunk Group Reference Audit.

Figure 11 Machine and trunk group reference audit output

```
$ audit sim1ae10 ref
Thu Jun 24 15:09:24 1993
** WARNING ** Too much reference data received from sim1ae10.
     With control counts, only enough space for 541 counts.
sim1ae10:  no response for sim4e14ent03-001
sim1ae10:  no usage register for sim4e14ent03-004
sim1ae10:  no peg count register for sim4e12ent04-020
sim1ae10:  no overflow register for sim4e13ent05-031
        sim4e14ent03 -002 : added to active database schedule
        sim4e14ent03 -003 : added to active database schedule
        sim4e14ent03 -005 : added to active database schedule
        sim4e14ent03 -006 : added to active database schedule
        sim4e14ent03 -007 : added to active database schedule
        sim4e14ent03 -009 : added to active database schedule
        sim4e14ent03 -010 : added to active database schedule
        sim4e14ent03 -011 : added to active database schedule
        sim4e14ent03 -012 : added to active database schedule
        sim4e14ent03 -015 : added to active database schedule
        sim4e12ent04 -021 : added to active database schedule
        sim4e12ent04 -023 : added to active database schedule
        tgid : 150 - trunk group not found in database
$
```

Figure 12 illustrates possible additional messages for the Machine and Trunk Group Reference Audit.

Figure 12 Possible messages for machine and trunk group reference audit

```
Tue May 23 14:42:03 1995
wlanca0127k : 1AESS REFERENCE DATA AUDIT
  tgid : 9 - Received TMC = 3 (2 way in pc) never received TMC = 0 for
    this TG - trunk type unknown
  tgid : 9 - Received TMC = 3 (2 way in pc) for trunk type = 0 (type not
    sent) - TMC and trunk type inconsistent
  tgid : 9 - Received TMC = 4 (2 way out pc) never received TMC = 0 for
    this TG - trunk type unknown
  tgid : 9 - Received TMC = 4 (2 way out pc) and trunk type = 0 (type
    not sent)
  tgid : 15 - Received TMC = 1 (1 way pc) for trunk type = 4 (svc ckt) -
    TMC and trunk type inconsistent
  tgid : 15 - Received TMC = 2 (overflow) for trunk type = 4 (svc ckt) -
    TMC and trunk type inconsistent
  tgid : 23 - Received TMC = 3 (2 way in pc) never received TMC = 0 for
    this TG - trunk type unknown
  tgid : 23 - Received TMC = 3 (2 way in pc) for trunk type = 0 (type
    not sent) - TMC and trunk type inconsistent
  tgid : 23 - Received TMC = 4 (2 way out pc) never received TMC = 0 for
    this TG - trunk type unknown
  tgid : 23 - Received TMC = 4 (2 way out pc) and trunk type = 0 (type
    not sent)
  tgid : 25 - Received TMC = 3 (2 way in pc) never received TMC = 0 for
    this TG - trunk type unknown
  tgid : 25 - Received TMC = 3 (2 way in pc) for trunk type = 0 (type
    not sent) - TMC and trunk type inconsistent
  tgid : 25 - Received TMC = 4 (2 way out pc) never received TMC = 0 for
    this TG - trunk type unknown
  tgid : 25 - Received TMC = 4 (2 way out pc) and trunk type = 0 (type
    not sent)
  tgid : 80 - Received TMC = 3 (2 way in pc) never received TMC = 0 for
    this TG - trunk type unknown
  tgid : 80 - Received TMC = 3 (2 way in pc) for trunk type = 0 (type
    not sent) - TMC and trunk type inconsistent
  tgid : 80 - Received TMC = 4 (2 way out pc) never received TMC = 0 for
    this TG - trunk type unknown
  tgid : 80 - Received TMC = 4 (2 way out pc) and trunk type = 0 (type
    not sent)
$
```

Information for switch coordinators

Table 2-1 shows 1A *ESS* TG valid values used for troubleshooting.

Table 2-1 1A ESS TG valid values

TRUNK TYPE	DIRECTION	END	PC	OFL	USG	IPC
1	out	A	tmc1	tmc2	tmc0	
(1-way)						
2	incoming	Z			tmc0	tmc1
3	2-way	A	tmc4	tmc2	tmc0	tmc3
(2-way)						
		Z	tmc4	tmc2	tmc0	tmc3
Unsupported:						
3	1-way out	A	tmc4	tmc2	tmc0	
(2-way used as 1-way)						
	1-way in	Z			tmc0	tmc3
2						

Modified trunk subgroup data audit (mtd)

Description

This audit is valid for the *4ESS* switch (generics 4e19 and later). This audit allows you to schedule an additional 255 MTD trunk groups beyond the 1024 or 2048 regular trunk groups. ***However***, this audit does not work like the typical Trunk group schedule audit (tgsched). There is no “audit type” for the MTD trunk groups. The MTD TGs can only be scheduled through the `tgsched` audit. Within the `tgsched` audit, the MTD trunk groups are cleared and scheduled at the switch and the MTD trunk group reference data is set up in the NTM database. These MTD requests require no action by the user since they are internal to the audit process. You can check the “/musr/log/aud_disc” or the” /musr/log/errors” files to verify that the `mtd` audit ran successfully or if there were any errors from the switch.

MTD trunk groups are scheduled internally by the audit process because of the design of the regular trunk groups in the *4ESS* switch. If a regular trunk group is unscheduled at the switch, the switch does not clear the Adjunct Head Cell (AHC) that is associated with that regular trunk group until a 15-minute boundary has occurred in the switch. Since the NTM database and the *4ESS* switch could be out of sync, 4 minutes have been added to this boundary condition. This means that the NTM audit process will send audit requests to the *4ESS* switch to schedule the MTD trunk groups and to request MTD trunk group reference data at 4, 19, 34, or 49 minutes after the hour. If a regular trunk group’s AHC is not removed, the trunk group cannot be scheduled as an MTD.

Sample output

When the `tgsched` audit is run:

- The message, “4ESS CLEARING MTD TRUNK GROUP SCHEDULE AUDIT” is displayed in the `tgsched` output.
- The typical `tgsched` output is displayed.
- ***No*** messages about the MTD trunk groups being scheduled are displayed. MTD audit requests are run internally and their success can be verified by checking the “/musr/log/aud_disc” or the “/musr/log/errors” files.

Notes to consider when scheduling MTD trunk groups

Be aware of the following when scheduling MTD trunk groups:

1. No controls can be set on MTD trunk groups.
2. If a regular trunk group has controls associated with it, and you want to make it an MTD trunk group, the controls must be removed before the trunk group can be scheduled as an MTD.

   If the control is not removed, the switch will refuse to schedule the trunk group as an MTD.

   There is no information provided by the switch to indicate that a control is set for a trunk group.
3. The 5-minute data associated with MTD trunk groups is a subset of the regular trunk group data.

   There are 6 fields of data associated with MTDs. The 7th data field, Egress Busy Crankback sent Peg Count, is always zero (0) and has no matching data field in the regular trunk group data.

   There are 10 trunk group fields and 9 flags that are associated with regular trunk group data.

   The 5-minute data fields that share the same names are:

   – Overflow Peg Count
   – Incoming Seizure Peg Counts
   – Occupancy Peg Count
   – Maintenance Busy Count

   The fields that have different names but the same values are shown in Table 3.
4. If a trunk group has never been scheduled, it must first be created and scheduled as a regular trunk group before it can be scheduled as an MTD trunk group. Failure to do this results in the switch refusing to schedule the trunk group as an MTD. The refusal happens most frequently after the switch is rebooted.
5. There is potential for not receiving three 5-minute data periods for MTD trunk groups as a result of the 15-minute boundary condition that exists in the switch.

   Reference: "Description" (p. 48)

Table

Table 3 shows fields that have different names but the same values

Table 3 MTD/regular trunk group data fields

MTD Trunk Group	Regular Trunk Group
Calls Rejected by CPE Peg Count	Crankback Count
Outgoing Attempt Peg Count	Attempts Peg Count

Scheduling MTD trunk groups

Instructions

Follow these steps to schedule MTD trunk groups:

1 Add the “sched+mtd” flags to the *options* field for each desired trunk group in the Trunk Group record base file (“/musr/rb/tg/<office_name>”).

2 Update the NTM database by performing a single office create.

3 Run the tgsched audit, which schedules the regular and MTD trunk groups in the *4ESS* switch.

END OF STEPS

OCL Dialed telephone number audit (ocldtn)

Description

The NTM allows you to execute a manual ocldtn audit type on User-Defined Network Elements (UDNE) of switch type "nextone" and can be run using the “udaudit” (p. 2-23) command.

This audit has been created to support the Outbound Call Limiting (OCL) conrol; it gets the number of SIP INVITES allowed per period of time for a destination telephone number.

Feature restrictions

This audit type is only available with the purchase of Feature 455, “Support for NexTone Session Border Controller Outbound Call Limiting”.

Sample output

An example of the output for the ocldtn audit.

```
$ udaudit ohrsm51_1 ocldtn
        IP
Thu Oct 30 14:18:03 200x
ohrsm51_1 : ocldtn audit.
control type=ocldtn office=ohrsm51_1 act=mod devname=msx01
   dtn=9953124659 ratelimitpolicy=ratelimitpolicy_22
control type=ocldtn office=ohrsm51_1 act=mod devname=msx1 dtn=3456789
   ratelimitpolicy=policy1978
control type=ocldtn office=ohrsm51_1 act=mod devname=msx1 dtn=668293111
   ratelimitpolicy=policy1978
```

□

OCL Signaling Endpoint audit (oclendpnt)

Description

The NTM allows you to execute a manual ocldtn audit type on User-Defined Network Elements (UDNE) of switch type "nextone" and can be run using the “udaudit” (p. 2-23) command.

This audit has been created to support the Outbound Call Limiting (OCL) conrol and is an enhanced version of the “Signaling Endpoint audit (eplist)” (p. 2-70).

Feature restrictions

This audit type is only available with the purchase of Feature 455, “Support for NexTone Session Border Controller Outbound Call Limiting”.

Sample output

An example of the output for the oclendpnt audit.

```
$ udaudit ohrsm51_1 oclendpnt
        IP
Thu Oct 30 15:23:05 200x
ohrsm51_1 : oclendpnt audit.
control type= oclendpnt office=ohrsm51_1 act=mod devname=msx01
   endpnt=broadsoft_as:0 ratelimitpolicy=ratelimitpolicy_22
control type= oclendpnt office=ohrsm51_1 act=mod devname=msx1
   endpnt=broadsoft_as:0 ratelimitpolicy=policy1978
control type= oclendpnt office=ohrsm51_1 act=mod devname=msx1
   endpnt=broadsoft_as:0 ratelimitpolicy=policy1978
```

□

OCL Signaling Realm audit (oclrealm)

Description

The NTM allows you to execute a manual ocldtn audit type on User-Defined Network Elements (UDNE) of switch type "nextone" and can be run using the “udaudit” (p. 2-23) command.

This audit has been created to support the Outbound Call Limiting (OCL) conrol and is an enhanced version of the “Signaling Realm audit (rlmlist)” (p. 2-72). The audit allows you to obtain a realm name of the realm that would receive the SIP INVITE.

Feature restrictions

This audit type is only available with the purchase of Feature 455, “Support for NexTone Session Border Controller Outbound Call Limiting”.

Sample output

An example of the output for the oclrealm audit.

```
$ udaudit ohrsm51_1 oclendpnt
        IP
Thu Oct 30 15:23:05 200x
ohrsm51_1 : oclendpnt audit.
control type= oclendpnt office=ohrsm51_1 act=mod devname=msx01
   endpnt=broadsoft_as:0 ratelimitpolicy=ratelimitpolicy_22
control type= oclendpnt office=ohrsm51_1 act=mod devname=msx1
   endpnt=broadsoft_as:0 ratelimitpolicy=policy1978
control type= oclendpnt office=ohrsm51_1 act=mod devname=msx1
   endpnt=broadsoft_as:0 ratelimitpolicy=policy1978
```

□

Packet schedule audit (pktsched)

Description

The packet schedule audit schedules packets for data collection from *5ESS*, 7R/E, *DMS*, *EWSD*, LSSGR, and SCSN switches. The scheduled packets are specified in the NTM record base.

If the packet number fails to be scheduled, only the packet schedule output is listed. The output contains the office name followed by the packet number and name that could not be scheduled.

Sample output

Figure 13 shows the messages that would appear if all packets were successfully scheduled.

Figure 13 Packet schedule audit output

```
$ audit dba5e6ent13+simdms24+1sgr+ewd pktsched
  IP
Thu Jun 24 15:09:24 1993
dba5e6ent13 : 5ESS PACKET SCHEDULE
simdms24 : DMS 100 PACKET SCHEDULE$
```

Figure 14 illustrates the output for an unsuccessful packet schedule audit. The message indicates that not all requested packets were successfully scheduled. If the switch is unable to schedule packets; contact customer support.

Figure 14 pktsched audit output (5ESS) — unsuccessful

```
$ audit snclca12ds0 pktsched
  IP
Thu Jun 24 15:09:24 1993
Packet 5, critical service ckts : could not schedule
Packet 1, delayed readiness     : could not schedule
Packet 3, call direction        : could not schedule
Packet 11, processor occupancy  : could not schedule$
```

References

See the “Office File” (p. 40) in the *Record Base Administration Guide* for more information on scheduling packets in the NTM record base

> **Important!** For LLSGR type switches, there are two conditions that will result in ***no*** 5-minute Trunk Group surveillance data being received in NTM due to interface incompatibility associated with scheduling 250 or 1024 trunk groups:

- NTM has requested the switch to schedule 250 trunk groups (Packet 17) but the switch is set up to schedule 1024 trunk groups (Packet 20)
- NTM has requested the switch to schedule 1024 trunk groups (Packet 20) but the switch is set up to schedule 250 trunk groups (Packet 17)

If there is no 5-minute data from the switch, see Chapter 15, “Surveillance Transition to Additional Trunk Groups” in the *System Administration Guide*.

□

Preprogram trunk group control audit (pp)

Description

This audit requests a list of active preprogram trunk group controls from the 1A *ESS* switch. This list is used to update the NTM database. Any controls deleted from or added to the NTM database appear in this audit output.

Sample output

Figure 15 illustrates the output of the Preprogram Trunk Group Control Audit output.

Figure 15 Preprogram trunk group control audit output

```
$ audit sim1ae10 pp
  IP
Thu Jun 24 15:09:24 1993
sim1ae10 : 1AESS PREPROGRAM CONTROL AUDIT
rr    from=sim1ae10 tg=sim4e14ent03-002 act=del
        hunt=spray alt=50 dir=50 pp=2 type=orr
        vias=sim4e14ent03-003+sim4e14ent03-004+sim4e14ent03-005
canf  from=sim1ae10 tg=sim4e14ent03-006 act=add
        alt=75 dir=100 annc=nca pp=4
rr    from=sim1ae10 tg=sim4e14ent03-009 act=add
        hunt=reg alt=50 dir=50 pp=6 type=irr
        vias=sim4e12ent04-021
$
```

□

Preprogram trunk group control list audit (pplist)

Description

This audit is used to update the NTM database with 1A *ESS* preprogram trunk group controls. The 1A *ESS* switches allow up to 63 “preprograms” to be defined in advance of their activation. When it is necessary to activate a preprogram, only the preprogram number is needed. For commonly used controls, this eliminates the need to reenter the same control.

There are four types of trunk group controls that can be defined as preprograms:

- CANT (Cancel To)
- CANF (Cancel From)
- SKIP
- RR (Reroute — regular and immediate)

NTM maintains an up-to-date list of preprograms per 1A *ESS*. If any changes occur in the 1A *ESS* switch preprograms, this audit runs automatically (it can also be run manually at any time). If any differences are detected between the NTM and the switch databases, NTM's database is updated. The audit output lists any preprograms deleted from or added to the database.

Sample output

Figure 16 illustrates the output of the Preprogram Trunk Group Control List.

Figure 16 Preprogram trunk group control list output

```
$ audit sim1ae10 pplist
  IP
Thu Jun 24 15:09:24 1993
sim1ae10 : 1AESS PREPROGRAM LIST AUDIT
cant  from=sim1ae10 tg=sim4e14ent03-003 act=add
        alt=50 dir=50 annc=nca pp=1
rr    from=sim1ae10 tg=sim4e14ent03-002 act=add
        hunt=spray alt=50 dir=50 pp=2 type=orr
        vias=sim4e14ent03-003+sim4e14ent03-004+sim4e14ent03-005
rr    from=sim1ae10 tg=sim4e14ent03-011 act=add
        hunt=spray alt=50 dir=50 pp=3 type=irr
        vias=sim4e14ent03-005+sim4e14ent03-006+sim4e14ent03-007
canf  from=sim1ae10 tg=sim4e14ent03-006 act=add
        alt=75 dir=100 annc=nca pp=4
skip  from=sim1ae10 tg=sim4e14ent03-007 act=add
        alt=50 dir=75 pp=5
rr    from=sim1ae10 tg=sim4e14ent03-009 act=add
```

```
        hunt=reg alt=50 dir=50 pp=6 type=irr
        vias=sim4e12ent04-021
canf  from=sim1ae10 tg=sim4e14ent03-010 act=add
        alt=75 dir=75 annc=nca pp=7
$
```

Possible errors

The following error message will be printed out by the `audit` command if a tgid was not found in the NTM database:

`<switch> tgid : <tgid> - trunk group not found in database`

This error could be caused by an incorrect keyword, incorrect spelling of a keyword, etc. Corrective actions include making these changes to the record base, doing a `create` and `installdb` of the database, and rerunning the audit.

□

Ratelimit control audit (ratelimit)

Description

The NTM allows users to execute manual ratelimit control audit type on Network Elemens of switchtype “nextone”. The ratelimit control audit checks for discrepancies in the ratelimit controls. Any differences are used to update the NTM database. You can execute an ratelimit audit on individual nextone elements, nextone elements in a network element set, or all nextone elements.

Feature restrictions

This audit type is only available with the purchase of Feature 438, “Support for NexTone Session Border Controller”.

Sample output

Figure 17 is an example of the output for the ratelimit audit.

Figure 17 ratelimit control audit output

```
$ udaudit tstnxt_1 ratelimit
  IP
Wed Sep 26 08:32:37 2007
tstnxt_1 : ratelimitaudit.
ratelimitnullcontrol type=ratelimit office=tstnxt_1 act=mod ulog=20
   limburst=10 devname=MSx1 part=admin thresh=114 name=def_ep_i_10
   limrate=20 layer=ip mod=input limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=60 limburst=20
   devname=MSx1 part=admin thresh=0 name=def_ep_i_100 limrate=100
   layer=ip mod=input limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=5 limburst=5
   devname=MSx1 part=admin thresh=5 name=def_ep_i_1000 limrate=5
   layer=ip mod=input limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=30 limburst=20
   devname=MSx1 part=admin thresh=113 name=def_ep_i_200 limrate=200
   layer=ip mod=input limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=60 limburst=20
   devname=MSx1 part=admin thresh=0 name=def_ep_i_2000 limrate=2000
   layer=ip mod=input limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=60 limburst=20
   devname=MSx1 part=admin thresh=0 name=def_ep_i_25 limrate=25 layer=ip
   mod=input limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=60 limburst=20
   devname=MSx1 part=admin thresh=0 name=def_ep_i_50 limrate=50 layer=ip
   mod=input limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=60 limburst=20
   devname=MSx1 part=admin thresh=113 name=def_ep_i_500 limrate=500
   layer=ip mod=input limtype=ep
```

```
control type=ratelimit office=tstnxt_1 act=mod ulog=60 limburst=-1
   devname=MSx1 part=admin thresh=0 name=def_ep_i_max limrate=-1
   layer=ip mod=input limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=0 limburst=20
   devname=MSx1 part=admin thresh=0 name=def_ep_o_10 limrate=10 layer=ip
   mod=output limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=0 limburst=20
   devname=MSx1 part=admin thresh=0 name=def_ep_o_100 limrate=100
   layer=ip mod=output limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=0 limburst=20
   devname=MSx1 part=admin thresh=0 name=def_ep_o_1000 limrate=1000
   layer=ip mod=output limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=0 limburst=20
   devname=MSx1 part=admin thresh=0 name=def_ep_o_200 limrate=200
   layer=ip mod=output limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=0 limburst=20
   devname=MSx1 part=admin thresh=0 name=def_ep_o_2000 limrate=2000
   layer=ip mod=output limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=333 limburst=21
   devname=MSx1 part=admin thresh=222 name=def_ep_o_25 limrate=25
   layer=ip mod=output limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=0 limburst=20
   devname=MSx1 part=admin thresh=0 name=def_ep_o_50 limrate=50 layer=ip
   mod=output limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=0 limburst=20
   devname=MSx1 part=admin thresh=0 name=def_ep_o_500 limrate=500
   layer=ip mod=output limtype=ep
control type=ratelimit office=tstnxt_1 act=mod ulog=0 limburst=-1
   devname=MSx1 part=admin thresh=0 name=def_ep_o_max limrate=-1
   layer=ip mod=output limtype=ep
```

□

Remote switching module audit (rsm)

Description

The remote switching module audit requests information related to any out-of-service or unreachable RSMs (Remote Switching Modules) and updates the NTM database with this information. You can manually initiate this audit or it will be automatically triggered by a discrete.

The `rsm` audit is sent to the switch via the control channel. Therefore, to run this audit, you must activate the switch (and the related DCC) for controls.

Sample output

Figure 18 illustrates the output for the `rsm` audit.

Figure 18 Remote switching module audit output

```
$ audit dba5e6ent13 rsm
  IP
Thu Jun 24 15:09:24 1993
dba5e6ent13 : 5ESS REMOTE SWITCHING MODULE AUDIT
        rsm 15:  rsm now available
        rsm  4:  rsm unable to switch
        rsm  7:  rsm out of contact
$
```

This sample shows those RSMs that have become unreachable. It also shows RSMs that have been unreachable and are now available. (RSMs for which communications have been reestablished appear first in the list.)

□

Reroute audit (rr)

Description

The reroute audit types are:

- Regular Reroute
- RDBI (Routing Data Block Index) Reroute
- Code Reroute
- Office Parameters Data

These reroute audits check for discrepancies in reroute control information and update the NTM database. You can request this audit manually, or it will be triggered automatically by a discrete.

Sample output

Figure 19 illustrates the output for a `reroute` audit.

Figure 19 Reroute audit output

```
$ audit sim4e13ent01 rr
  IP
Thu Jun 24 15:09:24 1993
sim4e13ent01 : 4ESS REROUTE AUDIT
rr    from=sim4e13ent01 tg=sim4e14ent03-005 act=del
        rrtype=reg tfc=dar rout=pct
        alt=50 dir=50 type=orr options=htr+nhr
        vias=sim4e14ent03-004 vrto=no domains=sdn
rr    from=sim4e13ent01 tg=sim4e14ent03-012 act=add
        rrtype=reg tfc=alt rout=pct
        alt=87 type=orr options=piit+htr
        vias=sim4e14ent03-010+sim4e14ent03-011+sim4e14ent03-012
        vrto=no+no+no domains=all
rr    from=sim4e13ent01 tg=sim4e13ent01-116 act=add
        rrtype=reg tfc=dar rout=pct
        alt=87 dir=87 type=orr options=htr+nhr
        vias=sim4e13ent01-105+sim4e13ent01-092+sim4e13ent01-244+
           sim4e13ent01-096+sim4e13ent01-131+sim4e13ent01-165+sim4e13ent01-138
        vrto=no+no+no+no+no+no+no domains=pots
sim4e13ent01 : 4ESS CODE REROUTE AUDIT
rr    from=sim4e13ent01 tg=sim4e12ent04-029 act=del
        rrtype=code tfc=alt rout=pct
        alt=75 type=irr options=cicr+htr+nhr
        vias=sim4e13ent05-031+sim4e13ent05-037+sim4e13ent05-039
        vrto=no+no+no domains=pots+orig+ssp+dedp
        codes=800891+123456+654321+987654+456789
```

```
rr    from=sim4e13ent01 tg=sim4e12ent04-021 act=add
        rrtype=code tfc=dar rout=rate
        type=irr options=apr+htr+nhr
        vias=sim4e13ent05-031+sim4e13ent05-036+sim4e13ent05-036+
           sim4e12ent06-041+sim4e12ent06-042+sim4e13ent07-051+sim4e12ent08-060
        vrto=no+no+no+no+no+no+no domains=sdn
        rate=all+1.5+2.5+4+6+10+30
        codes=111111+222222+333333+444444+555555+666666+777777+888888+
            999999+123123+234234+345345+890890+911911+900900+800800
rr    from=sim4e13ent01 tg=sim4e12ent04-025 act=add
        rrtype=code tfc=alt rout=pct
        alt=50 type=orr options=htr+nhr
        vias=sim4e13ent05-031+sim4e13ent05-036+sim4e13ent05-036
        vrto=no+no+no domains=sdn
        codes=111111+222222+333333+444444+890555+911666+900777+800888
sim4e13ent01 : : 4ESS RDBI REROUTE AUDIT
rr    from=sim4e13ent01 tg=sim4e14ent03-015 act=add
        rrtype=rdb tfc=dar rout=pct
        alt=37 dir=37 type=orr options=htr+nhr
        vias=sim4e12ent04-023+lab4e13ent17-075+lab4e12ent18-081+
           lab4e12ent18-082+lab4e12ent18-080+lab4e13ent17-072+lab4e13ent17-077
        vrto=no+no+no+no+no+no+no domains=ssp+ssp+ssp+ssp+ssp
        rdbi=12+13+14+15+16+17+18+19+
           111+211+311+411+511+611+711+811
sim4e13ent01 : BLK 35 : 4ESS OFFICE PARAMETERS DATA
     deleted dom:  30 intl:  30
     added   dom: 300 intl:  30
$
```

□

SBC audit (sbclist)

Description

The NTM allows users to execute manual sbclist audit type on Network Elemens of switchtype “nextone”. This audit gets the SBC list from the nextone Network Element and writes it into the file. You can execute an sbclist audit on individual nextone elements, nextone elements in a network element set, or all nextone elements.

Feature restrictions

This audit type is only available with the purchase of Feature 438, “Support for NexTone Session Border Controller”.

Office files created by sbclist

Current (*office*.c) and Old (*office*.o)

The `sbclist` audit creates a file called *office*.c (for ***current***) in the “/musr/list/sbclist/*clli*” directory. This file lists all SBCs currently in the RSM switch database. Each time this audit runs, the previous *office*.c file becomes an *office*.o file (for ***old*** file) and is placed in the directory called “/musr/list/sbclist/*clli.o*” As a result, a new *office*.c file is created and remains in the “/musr/list/sbclist/*clli*” directory.

New (*office*.n) and Remove (*office*.r)

The audit compares the list of SBC data currently defined in the RSM switch with those defined in the NTM record base (“/musr/rb/uddm/*sbc_office*”). Any SBC data found in the switch but not in NTM are placed in the file “*/musr/list/sbclist/clli/office*.n” (for ***new*** file). You can add these SBC data to the NTM record base if desired.

Any SBC data found in the NTM record base (“/musr/rb/uddm/*sbc_office*”) but not in the RSM switch are placed in the file “/musr/list/sbclist/*clli*/*office*.r” (for ***remove***). These SBC data either no longer belong in the record base or they need to be updated. Any SBC data listed in “/musr/list/sbclist/*clli*/*office*.r” file but not in “/musr/list/sbclist/*clli/office*.n” can be removed from the NTM record base if desired. If a SBC data is listed in both files, you should update the record base accordingly.

CAUTION

If you do not save existing .n and .r files before running this command, they will be deleted.

Extra (`office`.x)

If there are SBC data defined in the switch that you do not want to add to your record base, you can place them in "/musr/list/sbclist/`clli`/`office`.x" files (for ***extra***). This file tells the `sbclist` audit to ignore these SBC data when creating the "/musr/list/sbclist/`clli`/`office`.n" file. The SBC data in the `office`.x files will be considered part of NTM during the comparison and will not be listed in the "/musr/list/sbclist/`clli`/`office`.n" file.

Sample output

Figure 20 is an example of the output for the sbclist audit.

Figure 20 sbclist audit output

```
$ udaudit tstnxt_1 sbclist
        IP
Wed Sep 26 08:22:27 2007
tstnxt_1 : sbclistaudit.
sbclist 2 objects returned
$ ll
total 12
-rw-r--r--   1 nmadm      snm             46 Sep 26 08:22 tstnxt_1.c
-rw-r--r--   1 nmadm      snm             46 Sep 26 08:22 tstnxt_1.n
-rw-r--r--   1 nmadm      snm             46 Sep 25 20:31 tstnxt_1.o
-rw-r--r--   1 nmadm      snm              0 Sep 26 08:22 tstnxt_1.r
$ cat tstnxt_1.n
sbcname=MSx1, sbcid=1;
sbcname=MSx2, sbcid=2;
```

□

Selective incoming load control audit (silc)

Description

This audit checks for discrepancies between the trunk groups on which SILC is enabled in the *5ESS*, 7R/E ivtoa3 generic, or 1A *ESS* switches and trunk groups on which SILC is enabled in the NTM database. For the *GTD-5* switch, this audit is run as part of the Trunk group control audit (tgc).

Levels of congestion blocking percentages (MC1 [Machine Congestion Level 1] or MC2 [Machine Congestion Level 2]) of the office are also checked by the `silc` audit. Each trunk group is in either an “on” or “off” state on the silc list. An indicator is set depending on the trunk group state. The following are updated in the NTM database to match the switch database:

- silc list
- MC1 percentage
- MC2 percentage

Sample output

Figure 21 illustrates the output for the `silc` audit.

Figure 21 Selective incoming load control audit output

```
$ audit dba5e6ent13+sim1ae10 silc
  IP
Thu Jun 24 15:09:24 1993
dba5e6ent13 : 5ESS SILC AUDIT
        office silc percentages   mc1 is 50%, mc2 is 75%.
                sim4e12ent04 -022 : added to silc list
                sim4e14ent03 -003 : deleted from silc list
                sim4e14ent03 -010 : deleted from silc list
                sim4e13ent05 -033 : deleted from silc list
sim1ae10 : 1AESS SILC AUDIT
        office silc percentages   mc1 is 25%, mc2 is 100%.
                sim4e13ent01 -098 : added to silc list
                sim4e13ent01 -099 : added to silc list
                sim4e13ent01 -096 : added to silc list
                sim4e12ent08 -063 : deleted from silc list
$
```

The sample in Figure 21 shows the two machine count threshold values MC1 and MC2. These are followed by those trunk groups for which SILC has been turned on or off.

Important! Additions to the silc list always appear first.

□

Signaling link set audit (sls)

Description

The Signaling Link Set (SLS) audit provides a signaling reference from the Signaling Link Set ID (SLSID) to the following:

- Signaling Link Set Number (SLSNUM)
- Global Signaling Module Number (GSM)
- Signaling Type

The SLS audit checks for discrepancies between the signaling reference information in the switch and the signaling reference information in the NTM database. The audit updates the signaling reference information in NTM so that it matches the switch database.

The SLS audit lists only the discrepancies between the switch and the NMS database. The NTM database is then updated in the following ways:

- Signaling Link Sets that exist in the switch database, but not in the NTM database, are added to the NTM database.
- Signaling Link Sets that do not exist in the switch database, but exist in the NTM database, are deleted from the NTM database.
- Signaling Link Sets whose signaling reference information in the switch and NTM databases is different, are first deleted and then added to the NTM database.

The output of the SLS audit consists of the following information:

- Signaling link set numbers
- Global switching module numbers
- Signaling type
- 3-part adjacent point code (network_indicator:network_cluster:cluster_member)
- Signaling link set ID
- Notification as to whether the signaling link set was added to or deleted from the NTM database
- Global switching module index

Feature restrictions

This audit is available for *5ESS and* 7R/E offices only.

Sample output

Figure 22 illustrates the output for the `sls` audit.

Figure 22 Signaling link set audit output

```
$ audit all sls
  IP
SLSNUM=1  GSM=3  PROTOCOL=0
   APC=3:1:2  DEST_ENTITY=tstna014t
   LO_NUM_LINKS=3 HI_NUM_LINKS=2 - deleted from active database schedule
   SLSNUM=1  GSM=2  PROTOCOL=0
   APC=3:1:4  DEST_ENTITY=tstucs14t
   LO_NUM_LINKS=3 HI_NUM_LINKS=2 - deleted from active database schedule
   SLSNUM=1  GSM=1  PROTOCOL=0
   APC=4:1:3  DEST_ENTITY=tstna016t
   LO_NUM_LINKS=3 HI_NUM_LINKS=2 - deleted from active database schedule
   SLSNUM=1  GSM=1  PROTOCOL=0
   APC=4:1:3  DEST_ENTITY=tstna016t
   LO_NUM_LINKS=4 HI_NUM_LINKS=2 - added to active database schedule
   SLSNUM=1  GSM=2  PROTOCOL=0
   APC=3:1:4  DEST_ENTITY=tstucs14t
   LO_NUM_LINKS=3 HI_NUM_LINKS=5 - added to active database schedule
   SLSNUM=1  GSM=3  PROTOCOL=0
   APC=3:1:2  DEST_ENTITY=tstna014t
   LO_NUM_LINKS=2 HI_NUM_LINKS=6 - added to active database schedule
$
```

□

Signaling Endpoint audit (eplist)

Description

The NTM allows users to execute manual eplist audit type on Network Elemens of switchtype "nextone". This audit gets the Signaling Endpoints list from the nextone Network Element and writes it into the file. You can execute an eplist audit on individual nextone elements, nextone elements in a network element set, or all nextone elements.

Feature restrictions

This audit type is only available with the purchase of Feature 438, "Support for NexTone Session Border Controller".

Office files created by eplist

Current (*office*.c) and Old (*office*.o)

The `eplist` audit creates a file called *office*.c (for ***current***) in the "/musr/list/eplist/*clli*" directory. This file lists all endpoints currently defined in the switch database. Each time this audit runs, the previous *office*.c file becomes an *office*.o file (for ***old*** file) and is placed in the directory called "/musr/list/eplist/*clli*" As a result, a new *office*.c file is created and remains in the "/musr/list/eplist/*clli*" directory.

New (*office*.n) and Remove (*office*.r)

The audit compares the list of endpoints currently defined in the RSM switch with those defined in the NTM record base ("/musr/rb/uddm/*signalendpnt_office*"). Any endpoints found in the switch but not in NTM are placed in the file "/musr/list/eplist/*clli*/*office*.n" (for ***new*** file). You can add these endpoints to the NTM record base if desired.

Any endpoints found in the NTM record base but not in the RSM switch are placed in the file "/musr/list/eplist/*clli*/*office*.r" (for ***remove***). These endpoints either no longer belong in the record base or they need to be updated. Any endpoints listed in "/musr/list/eplist/*clli*/*office*.r" file but not in "/musr/list/eplist/*clli*/*office*.n" can be removed from the NTM record base if desired. If a endpoint is listed in both files, you should update the record base accordingly.

CAUTION

If you do not save existing .n and .r files before running this command, they will be deleted.

Extra (*office*.x)

If there are endpoints defined in the RSM switch that you do not want to add to your record base, you can place them in "/musr/list/eplist/*clli*/*office*.x" files (for ***extra***). This file tells the `eplist` audit to ignore these endpoints when creating the "/musr/list/eplist/*clli*/*office*.n" file. The endpoints in the *office*.x files will be considered part of NTM during the comparison and will not be listed in the "/musr/list/eplist/*clli*/*office*.n" file.

Sample output

Figure 23 is an example of the output for the eplist audit.

Figure 23 eplist audit output

```
$ udaudit tstnxt_1 eplist
        IP
Wed Sep 26 08:28:33 2007
tstnxt_1 : eplistaudit.
eplist  2 objects returned
$ ll
total 12
-rw-r--r--   1 nmadm      snm            204 Sep 26 08:28 tstnxt_1.c
-rw-r--r--   1 nmadm      snm            204 Sep 26 08:28 tstnxt_1.n
-rw-r--r--   1 nmadm      snm            204 Sep 26 08:23 tstnxt_1.o
-rw-r--r--   1 nmadm      snm              0 Sep 26 08:28 tstnxt_1.r
$ cat tstnxt_1.n
sbcname=MSx2, endpoint=Broadsoft:0, endpointip=222.168.1.18,
   realm=Public, end
pointtype=sipproxy;
sbcname=MSx1, endpoint=Broadsoft_AS:0, endpointip=222.168.1.18,
   realm=Public1,
 endpointtype=sipproxy;
```

□

Signaling Realm audit (rlmlist)

Description

The NTM allows users to execute manual rlmlist audit type on Network Elemens of switchtype “nextone”. This audit gets the Realms list from the nextone Network Element and writes it into the file. You can execute rlmlist audit on individual nextone elements, nextone elements in a network element set, or all nextone elements.

Feature restrictions

This audit type is only available with the purchase of Feature 438, “Support for NexTone Session Border Controller”.

Office files created by rlmlist

Current (*office*.c) and Old (*office*.o)

The `rlmlist` audit creates a file called *office*.c (for ***current***) in the “/musr/list/rlmlist/*clli*” directory. This file lists all realms currently defined in the RSM switch database. Each time this audit runs, the previous *office*.c file becomes an *office*.o file (for ***old*** file) and is placed in the directory called “/musr/list/rlmlist/*clli*.” As a result, a new *office*.c file is created and remains in the “/musr/list/rlmlist/*clli*” directory.

New (*office*.n) and Remove (*office*.r)

The audit compares the list of realms currently defined in the RSM switch with those defined in the NTM record base (“/musr/rb/uddm/*signalrealm_office*”). Any realms found in the RSM switch but not in NTM are placed in the file “/musr/list/rlmlist/*clli*/*office*.n” (for ***new*** file). You can add these realms to the NTM record base if desired.

Any realms found in the NTM record base but not in the RSM switch are placed in the file “/musr/list/rlmlist/*clli*/*office*.r” (for ***remove***). These realms either no longer belong in the record base or they need to be updated. Any realm listed in “/musr/list/rlmlist/*clli*/*office*.r” file but not in “/musr/list/rlmlist/*clli*/*office*.n” can be removed from the NTM record base if desired. If a realm is listed in both files, you should update the record base accordingly.

CAUTION

If you do not save existing .n and .r files before running this command, they will be deleted.

Extra (*office*.x)

If there are realms defined in the RSM switch that you do not want to add to your record base, you can place them in "/musr/list/rlmlist/*clli*/*office*.x" files (for ***extra***). This file tells the `rlmlist` audit to ignore these realms when creating the "/musr/list/rlmlist/*clli*/*office*.n" file. The realms in the *office*.x files will be considered part of NTM during the comparison and will not be listed in the "/musr/list/rlmlist/*clli*/*office*.n" file.

Sample output

Figure 24 is an example of the output for the rlmlist audit.

Figure 24 rlmlist audit output

```
$ udaudit tstnxt_1 rlmlist
        IP
Wed Sep 26 08:27:05 2007
tstnxt_1 : rlmlistaudit.
rlmlist 4 objects returned
$ ll
total 12
-rw-r--r--   1 nmadm      snm            116 Sep 26 08:27 tstnxt_1.c
-rw-r--r--   1 nmadm      snm            116 Sep 26 08:27 tstnxt_1.n
-rw-r--r--   1 nmadm      snm            116 Sep 25 20:45 tstnxt_1.o
-rw-r--r--   1 nmadm      snm              0 Sep 26 08:27 tstnxt_1.r
$ cat tstnxt_1.n
sbcname=MSx2, realm=Private;
sbcname=MSx1, realm=Public1;
sbcname=MSx1, realm=Private1;
sbcname=MSx2, realm=Public;
```

□

Trunk group control audit (tgc)

Description

The Trunk Group Control Audit checks for discrepancies in the active manual trunk group controls and updates the NTM database.

For the *GTD-5* switch, this audit internally runs the Automatic trunk group control audit (atg) and Selective incoming load control audit (silc) audits in addition to the Manual Trunk Group Control audit.

For *4ESS*, those audits are done in Trunk group reference and control audit (tginfo).

The audit output consists of a list of controls. These controls can be any of the following:

- CANT (Cancel To)
- CANF (Cancel From)
- SKIP
- RR (Reroute)
- CRO (Cancel Reroute Overflow)
- CR (Circuit Reservation)

Any differences found by the audit are used to update the NTM database and the Network View displays. This output consists of the location name of the office followed by the controls added to or deleted from the NTM database for that office.

Sample output

Figure 25 and Figure 26 illustrate the output for the trunk group control audit.

Figure 25 is an example of the output for the `tgc` audit for non-*GTD-5* switch types. The example shows trunk group controls that have been deleted from the switch but are still reflected in the NTM database. It also lists any existing trunk group controls that are in the switch but not in the NTM database. (Deletions appear first, followed by additions.)

Figure 25 Trunk group control audit output

```
$ audit dba5e6ent13+sim1ae10+simdms24 tgc
 IP
Thu Jun 24 15:09:24 1993
dba5e6ent13 : 5ESS MANUAL TRUNK GROUP CONTROL AUDIT
rr from=dba5e6ent13 tg=sim4e14ent03-010 act=add
    hunt=spray alt=25 dir=12
    vias=sim4e13ent05-034+sim4e13ent05-035+sim4e13ent05-036+
     sim4e13ent05-037+sim4e13ent05-038+sim4e13ent05-039+sim4e12ent06-040
    vrto=yes+yes+yes+yes+yes+yes+yes
    i/x=inc domains=iddd+data+sdn+dev+ded
```

```
cant  from=dba5e6ent13 tg=sim4e14ent03-014 act=add
        annc=nca alt=37 dir=12 i/x=xcl domains=all
skip  from=dba5e6ent13 tg=sim4e14ent03-018 act=add
        alt=37 dir=25 i/x=xcl domains=all
canf  from=dba5e6ent13 tg=sim4e12ent04-029 act=add
        annc=nca alt=37 dir=25 i/x=xcl domains=all
cro    from=dba5e6ent13 tg=sim4e13ent05-033 act=add
sim1ae10 : 1AESS MANUAL TRUNK GROUP CONTROL AUDIT
rr    from=sim1ae10 tg=sim4e14ent03-001 act=del
        hunt=reg alt=50 dir=50 type=orr
        vias=sim4e14ent03-002
rr    from=sim1ae10 tg=sim4e12ent08-060 act=del
        hunt=reg alt=12 dir=12 type=orr
        vias=sim4e14ent03-011
canf  from=sim1ae10 tg=sim4e13ent01-092 act=add
        alt=50 dir=50 annc=nca
cant  from=sim1ae10 tg=sim4e13ent01-091 act=add
        alt=50 dir=25 annc=nca
skip  from=sim1ae10 tg=sim4e13ent01-090 act=add
        alt=100 dir=100
$
```

Figure 26 is an example of the output for the `tgc` audit for the *GTD-5* switch.

Figure 26 Trunk group control audit output for GTD-5 switch

```
$ audit eriepa007gt tgc
IP
Sat Feb 26 09:55:49 1994
eriepa007gt : GTD-5 MANUAL TRUNK GROUP CONTROL AUDIT
  tgid : 179 - trunk group not found in database
Sat Feb 26 09:55:49 1994
eriepa007gt : GTD-5 AUTOMATIC TRUNK GROUP CONTROL AUDIT
cr    from=eriepa007gt tg=???-ct act=del
        sk/cn=cn cat=c lvlp1=10 lvlp2=5
cr    from=eriepa007gt tg=anhmca0295t-ct act=add
        sk/cn=cn cat=c lvlp1=10 lvlp2=5
Sat Feb 26 09:55:49 1994
eriepa007gt : GTD-5 SILC AUDIT
  office silc percentages   mc1 is 100%, mc2 is 50%.
     gtd5extern -ts38 : added to silc list
     gtd5extern -ts29 : added to silc list
     anhmca0152t -cn : added to silc list
     anhmca01amd -fg : added to silc list
$
```

Possible errors

The following error message will be printed out by the `audit` command if a tgid was not found in the NTM database:

```
<switch> tgid : <tgid> - trunk group not found in database
```

This error could be caused by an incorrect keyword, incorrect spelling of a keyword, etc. Corrective actions include making these changes to the record base, doing a `create` and `installdb` of the database, and rerunning the audit.

□

Trunk group list audit (tglist)

Description

This audit checks for discrepancies in trunk group list information. The reported information varies by switch type.

- For each trunk group from the *4ESS* switch database, this audit reports:
  - Trunk group number
  - BTFN (Base Traffic Number) — The BTFN must be specified in the associated "/musr/rb/tg" file for proper operation of this audit.
  - *CLLI* codes

  The `tglist` audit first checks to see that all trunk group numbers match. For those that do match, the `tglist` audit checks to see that the *CLLI* codes and BFTNs match.
- For each trunk group from the 1A *ESS*, *5ESS* switch (5e8 and later), 7R/E, *DMS*, *EWSD*, LSSGR, and SCSN database, this audit reports:
  - *CLLI* codes
  - Trunk group number
- For each trunk group from the *GTD-5* database, this audit reports:
  - *CLLI* codes
  - Trunk group number
  - Number of 1-way incoming circuits
  - Number of 1-way outgoing circuits
  - Number of 2-way circuits
  - Signaling type

Office files created by tglist

Current (*office*.c) and Old (*office*.o)

The `tglist` audit creates a file called *office*.c (for ***current***) in the "/musr/tglst/*clli*" directory. This file lists all trunk subgroups currently in the switch database. Each time this audit runs, the previous *office*.c file becomes an *office*.o file (for ***old*** file) and is placed in the directory called "/musr/tglst/*clli*." As a result, a new *office*.c file is created and remains in the "/musr/tglst/*clli*" directory.

New (*office*.n) and Remove (*office*.r)

The audit compares the list of trunk groups currently defined in the switch with those defined in the NTM record base ("/musr/rb/tg/*office*"). Any trunk groups found in the switch but not in NTM are placed in the file "/musr/tglst/*clli*/*office*.n" (for ***new*** file). You can add these trunk groups to the NTM record base if desired.

Any trunk groups found in the NTM record base but not in the switch are placed in the file "/musr/tglst/*clli*/*office*.r" (for ***remove***). These trunk groups either no longer belong in the record base or they need to be updated. Any trunk group listed in "/musr/tglst/*clli*/*office*.r" file but not in "/musr/tglst/*clli*/*office*.n" can be removed from the NTM record base if desired. If a trunk group is listed in both files, you should update the record base accordingly.

⚠ **CAUTION**

If you do not save existing .n and .r files before running this command, they will be deleted.

Extra (*office*.x)

If there are trunk groups defined in the switch that you do not want to add to your record base, you can place them in "/musr/tglst/*clli*/*office*.x" files (for ***extra***). This file tells the `tglist` audit to ignore these trunk groups when creating the "/musr/tglst/*clli*/*office*.n" file. The trunk groups in the *office*.x files will be considered part of NTM during the comparison and will not be listed in the "/musr/tglst/*clli*/*office*.n" file.

Sample output

Table 4 provides a list of the sample output figures, for ease-of-use in navigating online documentation.

Table 4 tglist sample output figures

Switch Type	See Figure
4ESS	Figure 27, "4ESS switch trunk group list audit output" (p. 79)
1A *ESS*	Figure 28, "1A ESS switch trunk group list audit output" (p. 80)
5ESS	Figure 29, "5ESS switch trunk group list audit output" (p. 81)
DMS	Figure 30, "DMS switch trunk group list audit output" (p. 81)

Table 4 tglist sample output figures

Switch Type	See Figure
LSSGR	Figure 31, “LSSGR trunk group list audit output” (p. 82)
EWSD	Figure 32, “EWSD switch trunk group list audit output” (p. 83)
GTD-5	Figure 33, “GTD-5 switch trunk group list audit output” (p. 84)

Figure 27 is an example of output from the `tglist` audit for a *4ESS* switch.

Note that the output is subdivided into those trunk groups listed in the “/musr/tglst/*clli*/*office*.c,” “/musr/tglst/*clli*/*office*.n,” “/musr/tglst/*clli*/*office*.x,” and “/musr/tglst/*clli*/*office*.o” files.

Figure 27 4ESS switch trunk group list audit output

```
$ audit sim4e13ent01 tglist
  IP
Thu Jun 24 15:09:24 1993
sim4e13ent01 : BLK 43 : 4ESS TRUNK GROUP LIST AUDIT
          10 Trunk groups returned
$ ls -o /musr/tglst/sim4e13ent01/sim4e13ent01*
-rw-rw-r--   1 nmadm        429 Mar 16 15:55 /musr/tglst/sim4e13ent01/sim4e13ent01.c
-rw-rw-r--   1 nmadm        117 Mar 16 15:55 /musr/tglst/sim4e13ent01/sim4e13ent01.n
-rw-rw-r--   1 nmadm         78 Mar 16 15:55 /musr/tglst/sim4e13ent01/sim4e13ent01.r
-rw-rw-r--   1 nmadm         78 Mar 16 15:55 /musr/tglst/sim4e13ent01/sim4e13ent01.x

$ cat /musr/tglst/sim4e13ent01/sim4e13ent01.c
to=    artnvasm01, tgn=   3, btfn=   3;
to=    artnvasm01, tgn=   4, btfn=   4;
to=    artnvasm01, tgn=   5, btfn=   5;
to=    artnvasm01, tgn=   6, btfn=   6;
to=    artnvasm01, tgn=   7, btfn=   7;
to=    artnvasm01, tgn=   8, btfn=   8;
to=    artnvasm01, tgn=   9, btfn=   9;
to=    artnvasm01, tgn=  10, btfn=  10;
to=    artnvasm01, tgn=  11, btfn=  11;
to=    artnvasm01, tgn=  12, btfn=  12;

$ cat /musr/tglst/sim4e13ent01/sim4e13ent01.r
to=    artnvasm01, tgn=   1, btfn=   1;
to=    artnvasm01, tgn=   2, btfn=   2;

$ cat /musr/tglst/sim4e13ent01/sim4e13ent01.x
to=    artnvasm01, tgn=   5, btfn=   5;
to=    artnvasm01, tgn=   6, btfn=   6;
$
```

Figure 28 is an example of output from the trunk group list audit for a 1A *ESS* switch.

Figure 28 1A ESS switch trunk group list audit output

```
$ audit sim1ae10 tglist
  IP
Thu Jun 24 15:09:24 1993
sim1ae10 : 1AESS TRUNK GROUP LIST AUDIT
          21 Trunk groups returned
$ ls -o /musr/tglst/sim1ae10/sim1ae10*
-rw-r--r--   1 nmadm       1158 Apr  3 14:56 /musr/tglst/sim1ae10/sim1ae10.c
-rw-r--r--   1 nmadm        154 Apr  3 14:56 /musr/tglst/sim1ae10/sim1ae10.n
-rw-r--r--   1 nmadm         62 Apr  3 14:56 /musr/tglst/sim1ae10/sim1ae10.r
$ cat /musr/tglst/sim1ae10/sim1ae10.c
to=sim4e14ent03-002,  tgn=   2;
to=sim4e14ent03-003,  tgn=   3;
to=sim4e14ent03-005,  tgn=   5;
to=sim4e14ent03-006,  tgn=   6;
to=sim4e14ent03-007,  tgn=   7;
to=sim4e14ent03-009,  tgn=   9;
to=sim4e14ent03-010,  tgn=  10;
to=sim4e14ent03-011,  tgn=  11;
to=sim4e14ent03-012,  tgn=  12;
to=sim4e14ent03-015,  tgn=  15;
to=sim4e12ent04-021,  tgn=  21;
to=sim4e12ent04-023,  tgn=  23;
$ cat /musr/tglst/sim1ae10/sim1ae10.n
to=sim4e14ent03-005,  tgn=   5;
to=                   tgn= 153;
to=                   tgn= 159;
$  cat /musr/tglst/sim1ae10/sim1ae10.r
to=sim4e13ent01-093,  tgn=  93;
ls -o /musr/tglst/sim1ae10/sim1ae10*
-rw-r--r--   1 nmadm       1066 Apr  3 14:56 /musr/tglst/sim1ae10/sim1ae10.o
$ cat /musr/tglst/sim1ae10/sim1ae10.o
to=sim4e14ent03-002,  tgn=   2;
to=sim4e14ent03-003,  tgn=   3;
to=sim4e14ent03-006,  tgn=   6;
to=sim4e14ent03-007,  tgn=   7;
to=sim4e14ent03-009,  tgn=   9;
to=sim4e14ent03-010,  tgn=  10;
to=sim4e14ent03-011,  tgn=  11;
to=sim4e14ent03-012,  tgn=  12;
to=sim4e12ent04-023,  tgn=  23;
to=sim4e13ent01-093,  tgn=  93;
$
```

Figure 29 is an example of output from the trunk group list audit for a *5ESS* switch.

Figure 29 5ESS switch trunk group list audit output

```
$ audit sim5e9ent01 tglist
  IP
sim5e9ent01 : 5ESS TRUNK GROUP LIST AUDIT
          10 Trunk groups returned
$ ls -o /musr/tglst/sim5e9ent01/sim5e9ent01*
-rw-r--r--   1 nmadm        300 Apr  3 16:53 /musr/tglst/sim5e9ent01/sim5e9ent01.c
-rw-r--r--   1 nmadm         90 Apr  3 16:53 /musr/tglst/sim5e9ent01/sim5e9ent01.n
-rw-r--r--   1 nmadm          0 Apr  3 16:53 /musr/tglst/sim5e9ent01/sim5e9ent01.r
$ cat /musr/tglst/sim5e9ent01/sim5e9ent01.c
to=clmboh1103t-001,  tgn=0001;
to=clmboh1103t-002,  tgn=0002;
to=clmboh1103t-003,  tgn=0003;
to=clmboh1103t-004,  tgn=0004;
to=clmboh1103t-005,  tgn=0005;
to=clmboh1103t-006,  tgn=0006;
to=clmboh1103t-007,  tgn=0007;
to=clmboh1103t-008,  tgn=0008;
to=clmboh1103t-009,  tgn=0009;
to=nycmny54zsz-2h,   tgn=0010;

$ cat /musr/tglst/sim5e9ent01/sim5e9ent01.n
to=clmboh1103t-008,  tgn=0008;
to=clmboh1103t-009,  tgn=0009;
to=nycmny54zsz-2h,   tgn=0010;

$ cat /musr/tglst/sim5e9ent01/sim5e9ent01.r
$ cat /musr/tglst/sim5e9ent0/sim5e9ent01.o
to=clmboh1103t-001,  tgn=0001;
to=clmboh1103t-002,  tgn=0002;
to=clmboh1103t-003,  tgn=0003;
to=clmboh1103t-004,  tgn=0004;
to=clmboh1103t-005,  tgn=0005;
to=clmboh1103t-006,  tgn=0006;
to=clmboh1103t-007,  tgn=0007;
$
```

Figure 30 is an example of output from the trunk group list audit for a *DMS* switch.

Figure 30 DMS switch trunk group list audit output

```
$ audit simdms24 tglist
  IP
Thu Jun 24 15:09:24 1993
simdms24 : DMS 100 TRUNK GROUP LIST AUDIT
          12 Trunk groups returned
$ ls -o /musr/tglst/simdms24/simdms24*
-rw-r--r--   1 nmadm        300 Apr  3 16:53 /musr/tglst/simdms24/simdms24.c
-rw-r--r--   1 nmadm         90 Apr  3 16:53 /musr/tglst/simdms24/simdms24.n
```

```
-rw-r--r--   1 nmadm          0 Apr  3 16:53 /musr/tglst/simdms24/simdms24.r
-rw-r--r--   1 nmadm         78 Apr  3 16:53 /musr/tglst/simdms24/simdms24.x
$ cat /musr/tglst/simdms24/simdms24/simdms24.c
to=    artnvasm01, tgn=   1;
to=    artnvasm01, tgn=   2;
to=    artnvasm01, tgn=   3;
to=    artnvasm01, tgn=   4;
to=    artnvasm01, tgn=   5;
to=    artnvasm01, tgn=   6;
to=    artnvasm01, tgn=   7;
to=    artnvasm01, tgn=   8;
to=    artnvasm01, tgn=   9;
to=    artnvasm01, tgn=  10;
to=    artnvasm01, tgn=  11;
to=    artnvasm01, tgn=  12;

$ cat /musr/tglst/simdms24/simdms24/simdms24.n
to=    artnvasm01, tgn=  10;
to=    artnvasm01, tgn=  11;
to=    artnvasm01, tgn=  12;

$ cat /musr/tglst/simdms24/simdms24/simdms24.r

$ cat /musr/tglst/simdms24/simdms24/simdms24.x
to=    artnvasm01, tgn=   6;
to=    artnvasm01, tgn=   7;
$
```

Figure 31 is an example of output from the trunk group list audit for an LSSGR switch.

Figure 31 LSSGR trunk group list audit output

```
$ audit lssgr tglist
  IP
Thu Jun 24 15:09:24 1993
lssgr: LSSGR TRUNK GROUP LIST AUDIT
          12 Trunk groups returned
$ ls -o /musr/tglst/simdms24/simdms24*
-rw-r--r--   1 nmadm        300 Apr  3 16:53 /musr/tglst/simdms24/simdms24.c
-rw-r--r--   1 nmadm         90 Apr  3 16:53 /musr/tglst/simdms24/simdms24.n
-rw-r--r--   1 nmadm          0 Apr  3 16:53 /musr/tglst/simdms24/simdms24.r
-rw-r--r--   1 nmadm         78 Apr  3 16:53 /musr/tglst/simdms24/simdms24.x
$ cat /musr/tglst/simdms24/simdms24.c
to=    artnvasm01, tgn=   1;
to=    artnvasm01, tgn=   2;
to=    artnvasm01, tgn=   3;
to=    artnvasm01, tgn=   4;
to=    artnvasm01, tgn=   5;
to=    artnvasm01, tgn=   6;
to=    artnvasm01, tgn=   7;
```

```
to=    artnvasm01, tgn=   8;
to=    artnvasm01, tgn=   9;
to=    artnvasm01, tgn=  10;
to=    artnvasm01, tgn=  11;
to=    artnvasm01, tgn=  12;

$ cat /musr/tglst/simdms24/simdms24.n
to=    artnvasm01, tgn=  10;
to=    artnvasm01, tgn=  11;
to=    artnvasm01, tgn=  12;

$ cat /musr/tglst/simdms24/simdms24.r

$ cat /musr/tglst/simdms24/simdms24.x
to=    artnvasm01, tgn=   6;
to=    artnvasm01, tgn=   7;
$
```

Figure 32 is an example of output from the trunk group list audit for an *EWSD* switch.

Figure 32 EWSD switch trunk group list audit output

```
$ audit ewsd tglist
  IP
Thu Jun 24 15:09:24 1993
ewsd : EWSD TRUNK GROUP LIST AUDIT
          12 Trunk groups returned
$ ls -o /musr/tglst/simdms24/simdms24*
-rw-r--r--   1 nmadm        300 Apr  3 16:53 /musr/tglst/simdms24/simdms24.c
-rw-r--r--   1 nmadm         90 Apr  3 16:53 /musr/tglst/simdms24/simdms24.n
-rw-r--r--   1 nmadm          0 Apr  3 16:53 /musr/tglst/simdms24/simdms24.r
-rw-r--r--   1 nmadm         78 Apr  3 16:53 /musr/tglst/simdms24/simdms24.x
$ cat /musr/tglst/simdms24/simdms24.c
to=    artnvasm01, tgn=   1;
to=    artnvasm01, tgn=   2;
to=    artnvasm01, tgn=   3;
to=    artnvasm01, tgn=   4;
to=    artnvasm01, tgn=   5;
to=    artnvasm01, tgn=   6;
to=    artnvasm01, tgn=   7;
to=    artnvasm01, tgn=   8;
to=    artnvasm01, tgn=   9;
to=    artnvasm01, tgn=  10;
to=    artnvasm01, tgn=  11;
to=    artnvasm01, tgn=  12;

$ cat /musr/tglst/simdms24/simdms24.n
to=    artnvasm01, tgn=  10;
to=    artnvasm01, tgn=  11;
to=    artnvasm01, tgn=  12;
```

```
$ cat /musr/tglst/simdms24/simdms24.r

$ cat /musr/tglst/simdms24/simdms24.x
to=    artnvasm01, tgn=   6;
to=    artnvasm01, tgn=   7;
$
```

Figure 33 is an example of output from the trunk group list audit for a *GTD-5* switch.

Figure 33 GTD-5 switch trunk group list audit output

```
$ audit eriepa007gt tglist
IP
Sat Feb 26 09:54:26 1994
eriepa007gt : GTD-5 TRUNK GROUP LIST AUDIT
   256 Trunk groups returned
$
$ cat /musr/tglst/eriepa007gt/eriepa007gt.c
to=anhmca01ds0_ab , tgn=0001 n1wi=0000 n1wo=0000 n2w=0100 sig=     ss7;
to=anhmca01amd_ac , tgn=0002 n1wi=0000 n1wo=0000 n2w=0100 sig=     ss7;
to=anhmca01ds0_ad , tgn=0003 n1wi=0100 n1wo=0000 n2w=0000 sig=     ss7;
to=anhmca01amd_ae , tgn=0004 n1wi=0000 n1wo=0100 n2w=0000 sig=     ss7;
to=anhmca01ds0_af , tgn=0005 n1wi=0000 n1wo=0000 n2w=0100 sig=     ss7;
to=anhmca01amd_ag , tgn=0006 n1wi=0100 n1wo=0000 n2w=0000 sig=     ss7;
to=anhmca01ds0_ah , tgn=0007 n1wi=0000 n1wo=0100 n2w=0000 sig=     ss7;
to=anhmca01amd_ai , tgn=0008 n1wi=0000 n1wo=0000 n2w=0100 sig=     ss7;
to=anhmca01ds0_aj , tgn=0009 n1wi=0100 n1wo=0000 n2w=0000 sig=     ss7;
to=anhmca01amd_ak , tgn=0010 n1wi=0000 n1wo=0100 n2w=0000 sig=     ss7;

$ cat /musr/tglst/eriepa007gt/eriepa007gt.n
to=anhmca01amd_ac , tgn=0002 n1wi=0000 n1wo=0000 n2w=0100 sig=     ss7;
to=anhmca01ds0_ad , tgn=0003 n1wi=0100 n1wo=0000 n2w=0000 sig=     ss7;

$ cat /musr/tglst/eriepa007gt/eriepa007gt.r
to= anhmca01amd-ac, tgn=0002 n1wi=0000 n1wo=0000 n2w=0000 sig=     ss7;
to= anhmca01ds0-ad, tgn=0003 n1wi=0000 n1wo=0000 n2w=0000 sig=     ss7;

$ cat /musr/tglst/eriepa007gt/eriepa007gt.x
to=anhmca01amd_ae , tgn=0004 n1wi=0000 n1wo=0100 n2w=0000 sig=     ss7;
cbnmsc>
```

Possible errors

The following error message may appear when a `tglist` audit is run:

```
audit hstntx0154t tglist
 IP hstntx0154t : BLK 43 : 4ESS TRUNK GROUP LIST AUDIT
 Cannot modify file /musr/tglst/hstntx0154t.c : permission
 denied
 tglist audit (block 43) failed for hstntx0154t.
```

This message indicates that the tglist file cannot be modified by the audit because of incorrect file permissions. If necessary, you can modify the file permissions and rerun the audit, or consult your Record Base administrator.

References

See “Setting up trunk group files” (p. 19) in the *Record Base Administration Guide* for more information on how to add and remove trunk groups from the NTM record base.

□

Trunk group reference audit (tg)

Description

The trunk group reference audit checks for discrepancies in the trunk group reference data. Scheduled trunk groups are the only trunk groups included in the audit. The output consists of trunk groups being added to or deleted from the database schedule. If the audit detects a trunk group it does not recognize, it will output a message saying that the trunk group number is “not found.”

A change in trunk group schedule should only occur after a local network management action takes place at the switch.

For the *GTD-5* switch, this audit only updates the reference data for the number of circuits in a trunk group. The scheduling of trunk groups is performed by the Trunk group schedule audit (tgsched).

Sample output

Figure 34 illustrates the output for the trunk group reference audit for non-*GTD-5* switches. This sample shows those trunk groups that have just been added to the active database schedule. These trunk groups are followed by a list of trunk groups that were just removed from the active database schedule.

Figure 35 illustrates the output for the trunk group reference audit for the *GTD-5* switch.

Figure 34 Trunk group reference audit output — non-GTD-5

```
$ audit dba5e6ent13+simdms24 tg
  IP
Thu Jun 24 15:09:24 1993
dba5e6ent13 : 5ESS TRUNK GROUP REFERENCE AUDIT
        sim4e13ent05 -035 : added to active database schedule
        sim4e13ent05 -036 : added to active database schedule
        sim4e13ent05 -037 : added to active database schedule
        sim4e12ent06 -043 : added to active database schedule
        sim4e14ent03 -008 : deleted from active database schedule
simdms24 : DMS 100 TRUNK GROUP REFERENCE AUDIT
        sim4e14ent03 -001 : added to active database schedule
        sim4e14ent03 -002 : added to active database schedule
        sim4e14ent03 -003 : added to active database schedule
sim4e14ent03 -005 : added to active database schedule
        sim4e14ent03 -006 : added to active database schedule
        sim4e14ent03 -007 : added to active database schedule
sim4e14ent03 -009 : deleted from active database schedule
        sim4e14ent03 -010 : deleted from active database schedule
$
```

Figure 35 Trunk group reference audit output — GTD-5

```
$ audit eriepa007gt tg
  IP
Thu May 12 08:49:15 1994
eriepa007gt : GTD-5 TRUNK GROUP REFERENCE AUDIT
        tgid : 255 - trunk group not found in switch database
        tgid : 256 - trunk group not found in switch database
        tgid : 427 - trunk group not found in database
        tgid : 2001 - trunk group not found in database
$
```

□

Trunk group reference and control audit (tginfo)

Description

The trunk group reference and control audit checks for discrepancies in trunk group information. Trunk group information can change each time a network management action is performed at the *4ESS* switch (for example, adding a control to a particular trunk group). An alerting discrete is automatically set at the *4ESS* switch when the trunk group information changes.

When trunk group information changes, the audit runs automatically if the trunk group reference and control audit has not been previously inhibited. The trunk group reference information, manual trunk group controls, and automatic trunk group controls are all contained in the same *4ESS* switch block of information.

Sample output

Figure 36 illustrates the trunk group reference and control audit output.

Figure 36 Trunk group reference and control audit output

```
$ audit sim4e13ent01 tginfo
  IP
Thu Jun 24 15:09:24 1993
sim4e13ent01 : 4ESS TRUNK GROUP REFERENCE AUDIT
   sim4e14ent03 -005 : added to active database schedule
        sim4e14ent03 -006 : added to active database schedule
        sim4e14ent03 -007 : added to active database schedule
        sim4e14ent03 -009 : deleted from active database schedule
        sim4e14ent03 -010 : deleted from active database schedule
sim4e13ent01 : 4ESS MANUAL TRUNK GROUP CONTROL AUDIT
cant  from=sim4e13ent01 tg=sim4e14ent03-009 act=del
   annc=ssb rout=pct alt=75 htr=h-u
skip  from=sim4e13ent01 tg=sim4e14ent03-005 act=add
   rout=pct alt=50 htr=h-u
cro   from=sim4e13ent01 tg=sim4e14ent03-016 act=add
canf  from=sim4e13ent01 tg=sim4e14ent03-018 act=add
   annc=ea1 rout=pct alt=100 htr=h-u
sim4e13ent01 : 4ESS AUTOMATIC TRUNK GROUP CONTROL AUDIT
cr    from=sim4e13ent01 tg=sim4e14ent03-003 act=del
   ih/rs=rs sk/cn=sk cat=a l1=10 l2=5
acc   from=sim4e13ent01 tg=sim4e14ent03-013 act=add
   ih/rs=ih sk/cn=cn cat=b
cr    from=sim4e13ent01 tg=sim4e14ent03-015 act=add
   ih/rs=rs sk/cn=cn cat=a l1=15 l2=10
$
```

This sample shows output for trunk group reference data, manual trunk group control data, and automatic trunk group control data. You receive all three types of data by selecting the `tginfo` audit.

□

Trunk group schedule audit (tgsched)

Description

The trunk group schedule audit adds or removes trunk groups from the trunk group schedule for a switch. The schedule can contain up to the following number of trunk groups for these switch types:

- 1024 — *DMS*, SCSN, 4E17
- 2000 — *5ESS*, 7R/E
- 2047 — 4e18+

Switches send 5-minute data to the NTM database for trunk groups assigned in the record base. The 5-minute data is not sent if the trunk group schedule or trunk group reference data audit has not been previously run. The trunk group schedule allows only those trunk groups that have a schedule indicator set in the NTM database to be scheduled.

The trunk group schedule output consists of the audit type and the office name. These are followed by a list of trunk groups that could not be scheduled, if any. The Trunk group reference audit (tg) reports reference data for scheduled trunk groups only.

Notes

Please note the following:

- For the *GTD-5*, this audit automatically runs the Trunk group schedule list audit (tgschedlist) before and after running the `tgsched` audit.

  Reference: "Trunk group schedule audit for the GTD-5 switch" (p. 91)
- EADAS/TDMS limits scheduling to only 25 trunk groups at a time. This limitation causes the `tgsched` audit output to look as though a series of audits has taken place.
- The FEP/*GTD-5* limits scheduling to five trunk groups at a time. This limitation causes the `tgsched` audit output to look as though a series of audits has taken place.
- For the *4ESS*, this audit automatically runs the Trunk group reference and control audit (tginfo) before and after running the `tgsched` audit. With generic 4e19, the Modified trunk subgroup data audit (mtd) will be added to the `tgsched` audit.

Sample output

Figure 37 illustrates the output for a successful trunk group schedule audit. The sample shown in Figure 37 indicates the audit scheduled 88 trunk groups for the *DMS* 100/200 switch and 87 trunk groups for the *5ESS* switch.

Figure 37 Trunk group schedule audit output

```
$ audit simdms24+dba5e6ent13 tgsched
  IP
Thu Jun 24 15:09:24 1993
simdms24 : DMS 100 TRUNK GROUP SCHEDULE
        Clearing ALL trunk groups in simdms24

simdms24 : DMS 100 TRUNK GROUP SCHEDULE
        Adding next 25 trunk groups to simdms24 schedule
63 more to do
simdms24 : DMS 100 TRUNK GROUP SCHEDULE
        Adding next 25 trunk groups to simdms24 schedule
        38 more to do
simdms24 : DMS 100 TRUNK GROUP SCHEDULE
        Adding next 25 trunk groups to simdms24 schedule
        13 more to do
simdms24 : DMS 100 TRUNK GROUP SCHEDULE
        Adding next 13 trunk groups to simdms24 schedule
dba5e6ent13 : 5ESS TRUNK GROUP SCHEDULE
        Clearing ALL trunk groups in dba5e6ent13
dba5e6ent13 : 5ESS TRUNK GROUP SCHEDULE
        Adding next 25 trunk groups to dba5e6ent13 schedule
        62 more to do
dba5e6ent13 : 5ESS TRUNK GROUP SCHEDULE
        Adding next 25 trunk groups to dba5e6ent13 schedule
        37 more to do
dba5e6ent13 : 5ESS TRUNK GROUP SCHEDULE
        Adding next 25 trunk groups to dba5e6ent13 schedule
        12 more to do
dba5e6ent13 : 5ESS TRUNK GROUP SCHEDULE
        Adding next 12 trunk groups to dba5e6ent13 schedule
$
```

Trunk group schedule audit for the *GTD-5* switch

For the *GTD-5* switch, the `tgsched` audit automatically runs the Trunk group schedule list audit (tgschedlist), which collects the current trunk group scheduling information from the switch and updates the NTM database, before and after running the `tgsched` audit. Therefore, `tgsched` output for the *GTD-5* switch looks slightly different.

> **Important!** The *GTD-5* switch may have trunks scheduled by craft at the switch. This can result in the scheduling of the same trunk group multiple times and a reduction of the total number of different trunk groups that can be scheduled. This can also cause the audit to fail.

Reference: See "Trunk group schedule list audit (tgschedlist)" (p. 93) for more information on `tgsched` and `tgschedlist` for the *GTD-5* switch and for an example of output for the combined `tgsched` and `tgschedlist` audit for the *GTD-5* switch.

Possible error messages

The following error messages may occur during a `tgsched` audit.

- The following two messages

  ```
  Switch indicated trunk group schedule list overflow
  ```

  and

  ```
  REPORT AUD TBL 817 nms <system_name> AUDSRV: switch indicated trunk
  group schedule list overflow
  ```

  indicate that NTM could not schedule all trunk groups that were requested. This is probably because craft at the switch requested trunk groups independently of NTM. The switch has notified NTM that it has overflowed the table of scheduled trunk groups.
- The message

  ```
  <clli> could not schedule active database schedule
  ```

  indicates that the CLLI code for the trunk groups was not scheduled.

Possible reasons for this message include trunk groups with no circuits and other failures such as trunk group schedule list overflow.

□

Trunk group schedule list audit (tgschedlist)

Description

The Trunk Group Schedule List Audit collects the current trunk group scheduling information from the *GTD-5* switch and updates the NTM database.

The Trunk Group Schedule List Audit is specific to the *GTD-5* switch. It is not a true audit “type” as it cannot be run separately. It runs automatically when a Trunk group schedule audit (tgsched) is requested for a *GTD-5* switch. Since the *GTD-5* does not support discrete-triggered audits, the `tgschedlist` audit is run both before and after the Trunk Group Schedule audit.

Sample output

Figure 38 illustrates the combined output for the Trunk Group Schedule and Trunk Group Schedule List audits.

Figure 38 Trunk group schedule list audit output

```
$ audit eriepa007gt tgsched
IP
Sat Feb 26 10:00:36 1994
eriepa007gt : GTD-5 TRUNK GROUP SCHEDULE LIST
  anhmca01amd-se : added to active database schedule
  anhmca01ds0-ph : added to active database schedule
  anhmca0152t-da : added to active database schedule
  anhmca0152t-cn : added to active database schedule
  anhmca0152t-nc : added to active database schedule
  anhmca0177k-ph : added to active database schedule
  anhmca0295t-af : added to active database schedule
  anhmca0295t-ct : added to active database schedule
  carrier prefix 22 added to active database schedule
  carrier prefix 88 added to active database schedule
  gtd5extern-tst1 : added to active database schedule
  carrier prefix 1 added to active database schedule
  carrier prefix 1 added to active database schedule
  carrier prefix 1 added to active database schedule
  gtd5extern-tst9 : added to active database schedule
  gtd5extern-ts10 : added to active database schedule
  gtd5extern-ts11 : added to active database schedule
  gtd5extern-ts12 : added to active database schedule
  gtd5extern-ts13 : added to active database schedule
  gtd5extern-ts14 : added to active database schedule
  gtd5extern-ts15 : added to active database schedule
  gtd5extern-ts16 : added to active database schedule
  gtd5extern-ts17 : added to active database schedule
  gtd5extern-ts18 : added to active database schedule
  gtd5extern-ts19 : added to active database schedule
  gtd5extern-ts20 : added to active database schedule
  gtd5extern-ts21 : added to active database schedule
  gtd5extern-ts22 : added to active database schedule
  gtd5extern-ts23 : added to active database schedule
  gtd5extern-ts24 : added to active database schedule
  gtd5extern-ts25 : added to active database schedule
  gtd5extern-ts26 : added to active database schedule
  gtd5extern-ts27 : added to active database schedule
  gtd5extern-ts28 : added to active database schedule
  gtd5extern-ts29 : added to active database schedule
  gtd5extern-ts30 : added to active database schedule
  gtd5extern-ts31 : added to active database schedule
  bkfdca12ds0-9990 : added to active database schedule
  ewsd10sim-ts33 : added to active database schedule
  gtd5extern-ts34 : added to active database schedule
  gtd5extern-ts35 : added to active database schedule
  gtd5extern-ts36 : added to active database schedule
  gtd5extern-ts37 : added to active database schedule
  gtd5extern-ts38 : added to active database schedule
  carrier prefix 39 added to active database schedule
Sat Feb 26 10:00:44 1994
eriepa007gt : GTD-5 TRUNK GROUP SCHEDULE
  Deleting next 5 trunk groups from eriepa007gt schedule
Sat Feb 26 10:00:47 1994
eriepa007gt : GTD-5 TRUNK GROUP SCHEDULE
  Deleting next 5 trunk groups from eriepa007gt schedule
Sat Feb 26 10:00:50 1994
eriepa007gt : GTD-5 TRUNK GROUP SCHEDULE
  Deleting next 5 trunk groups from eriepa007gt schedule
Sat Feb 26 10:00:53 1994
eriepa007gt : GTD-5 TRUNK GROUP SCHEDULE
  Deleting next 5 trunk groups from eriepa007gt schedule
Sat Feb 26 10:00:56 1994
eriepa007gt : GTD-5 TRUNK GROUP SCHEDULE
  Deleting next 5 trunk groups from eriepa007gt schedule
Sat Feb 26 10:00:59 1994
eriepa007gt : GTD-5 TRUNK GROUP SCHEDULE
  Deleting next 5 trunk groups from eriepa007gt schedule
Sat Feb 26 10:01:07 1994
eriepa007gt : GTD-5 TRUNK GROUP SCHEDULE
  Deleting next 5 trunk groups from eriepa007gt schedule
Sat Feb 26 10:01:10 1994
eriepa007gt : GTD-5 TRUNK GROUP SCHEDULE
  Deleting next 2 trunk groups from eriepa007gt schedule
  2 more to do
Sat Feb 26 10:01:26 1994
eriepa007gt : GTD-5 TRUNK GROUP SCHEDULE LIST
```

```
$
```

□

Wsdl audit (wsdl)

Description

The NTM allows users to execute manual wsdl audit type on Network Elements of switchtype “nextone”. The wsdl audit gets the list of wsdl files from nextone Network Element, compares them with existing wsdl files and presents the list of differences. You can execute an wsdl audit on individual nextone elements, nextone elements in a network element set, or all nextone elements.

Feature restrictions

This audit type is only available with the purchase of Feature 438, “Support for NexTone Session Border Controller”.

Sample output

Figure 39 illustrates the output for the wsdl audit.

Figure 39 wsdl audit output

```
$ udaudit tstnxt_1 wsdl
   IP
Wed Sep 26 09:04:11 2007
tstnxt_1 : wsdlaudit.
wsdlReference file /musr/rb/udnei/soap/nextonersm/rsm5_0/sqlreport.wsdl is not
   readableComparing:
   /musr/swdata/udnei/nextonersm/wsdl.tstnxt_1.sqlreportxsd.response.xml,
   /musr/rb/udnei/soap/nextonersm/rsm5_0/sqlreport.xsd1c1,156
< <?xml version="1.0" encoding="UTF-8"?><wsdl:definitions
   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
   xmlns="http://www.nextone.com/ivms/wsdl"
   xmlns:ns="http://www.nextone.com/ivms/schema/"
   xmlns:ns1="http://www.nextone.com/ivms/schema/common"
   xmlns:nsr="http://www.nextone.com/ivms/schema/sqlreport"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
   xmlns:xs="http://www.w3.org/2001/XMLSchema" name="NexToneSqlReport"
   targetNamespace="http://www.nextone.com/ivms/wsdl"> <wsdl:documentation>Copyright
   2005, NexTone Communications</wsdl:documentation> <wsdl:types> <xs:schema
   xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
   elementFormDefault="qualified"
   targetNamespace="http://www.nextone.com/ivms/schema/wsdl"> <xs:import
   namespace="http://www.nextone.com/ivms/schema/sqlreport"
   schemaLocation="sqlreport.xsd"/> </xs:schema> </wsdl:types><wsdl:message
   name="sqlReportResponse"> <wsdl:part element="nsr:reply"
   name="sqlReportResponse"/> </wsdl:message> <wsdl:message name="sqlReportRequest">
   <wsdl:part element="nsr:get" name="sqlReportRequest"/> </wsdl:message>
   <wsdl:portType name="SqlReportPortType"> <wsdl:operation name="getSqlReport">
   <wsdl:input message="sqlReportRequest" name="reportRequest"/> <wsdl:output
   message="sqlReportResponse" name="reportResponse"/> </wsdl:operation>
   </wsdl:portType> <wsdl:binding name="SqlReportBinding" type="SqlReportPortType">
   <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
   <wsdl:operation name="getSqlReport"> <soap:operation style="document"/>
   <wsdl:input> <soap:body use="literal"/></wsdl:input> <wsdl:output> <soap:body
   use="literal"/> </wsdl:output></wsdl:operation> </wsdl:binding> <wsdl:service
   name="SqlReportService"> <wsdl:port binding="SqlReportBinding"
   name="SqlReportPort"> <soap:address
   location="https://RSM.lab:8443/rsm/ws/prov/sqlreport"/>
   </wsdl:port></wsdl:service> </wsdl:definitions>
---
> <?xml version="1.0" encoding="UTF-8"?>
> <xs:schema attributeFormDefault="unqualified" elementFormDefault="unqualified"
   targetNamespace="http://www.nextone.com/ivms/schema/sqlreport"
   xmlns="http://www.nextone.com/ivms/schema/sqlreport"
   xmlns:nsc="http://www.nextone.com/ivms/schema/common"
   xmlns:xs="http://www.w3.org/2001/XMLSchema">
>  <xs:annotation>
>     <xs:documentation>This schema contains the definition of XML elements used in
   SQL report web service. SQL report web service allows retrieval of reports from
   RSM by specifying the parts of sql in the request</xs:documentation>
>  </xs:annotation>
>  <xs:import namespace="http://www.nextone.com/ivms/schema/common"
   schemaLocation="common.xsd"/>
>  <xs:element name="reply">
>     <xs:annotation>
>        <xs:documentation>reply element is returned as a reponse to sql web service
   get request. Only of of error or data element is present in the
   reponse</xs:documentation>
>     </xs:annotation>
>     <xs:complexType>
>        <xs:sequence>
>           <xs:element minOccurs="0" name="ok" nillable="true" type="xs:string"/>
>           <xs:element minOccurs="0" name="error" nillable="true"
   type="nsc:errorType"/>
>           <xs:element minOccurs="0" name="data" nillable="true"
   type="dataReportType"/>
>        </xs:sequence>
>     </xs:complexType>
>  </xs:element>
```

□

3 BDR Commands

Overview

Purpose

This chapter lists and describes the NTM BDR commands. These commands are executed from the BDR host computer in order to facilitate BDR’s purpose of enabling a Network Management Center (NMC) and/or host computer to back up another NMC and/or host computer for the following situations:

- An NMC center fails but its NTM host computer does not. The personnel at a backup NMC take over the failed NMC's operations, using the failed center's NTM host.
- An NTM host fails but its NMC does not. The personnel at its NMC continue their operations, using their backup NTM host.
- An NTM host and its NMC fail. The personnel at a backup NMC take over the operations for the failed location, using their own NTM host.

Feature restrictions

BDR commands are only available with the BDR (Backup and Disaster Recovery) features: Feature 8, “Disaster Recovery (Duplex)” and Feature 40, “Enhanced Disaster Recovery”.

References

Chapter 12, “BDR Administration on a Host” in the *System Administration Guide*

Chapter 9, “Maintaining the Record Base with BDR” in the *Record Base Administration Guide*

Contents

This chapter discusses the following topics:

□

BDR error and information logs

Overview

The following error and information logs associated with the BDR feature can be found on the host computer:

- “/musr/log/bdrlog” — BDR errors and information
- “/musr/log/recreate.out|recreate.out1|recreate.out2” — output of the dayend `recreate` command

  The `recreate` command runs nightly. NTM preserves three nightly output files called “recreate.out” (most recent), “recreate.out1” and “recreate.out2”.

  Check “/musr/log/recreate.out” daily to see if any `dbtest` or `create` errors occurred during the `recreate` portion of the `dayend` procedure or if any audits need to be rerun.

  The message “Cannot communicate with data collector” may mean that the data collector was down when `recreate` was trying to run. In that case, run the `deact`, `create`, `act`, and `audit` commands manually.
- “/musr/log/aud_<hostname>” — information on audit offices between hosts
- “/musr/auds/rsp/<hostname>/<office>” — each office directory contains one raw data file of manual audit responses for each audit type
- “/musr/auds/tmp” — contains discrete-triggered audit data files
- “/musr/ctrl” — contains control data on which switches have had control changes since the last `dayend` ran

□

Command functions

Table

Table 1 lists each NTM BDR command and its function.

Table 1 BDR commands and functions

Command	Function
`bdr_act`	Activates BDR on the NTM host computer
`bdr_audit`	Audits all backup entities
`bdr_chgstat`	Changes or checks the status of global record base files on the host computer
`bdr_commit`	Synchronizes changes made to the global record base files
`bdr_deact`	Deactivates BDR on the NTM host computer
`bdr_switchbk`	Switches a host computer from takeover mode back to normal mode
`bdr_sync`	Synchronizes record base files on one host computer with those on another host computer
`bdr_takeover`	Allows a remote host computer to take over the functionality of a failed host computer
`sync_audit`	Synchronizes the data base controls that reside in the secondary subnetwork partitions on this NTM host

□

bdr_act

Description

The `bdr_act` (activate BDR) command is used on the host to:

- display the current status of the BDR (`active`, `not active`, or `undefined`)
- change the state of BDR to `active` on an NTM host. This will enable the application software on this host to perform backup to other hosts and vice versa.

Syntax

```
/nm/sys/bdr_act
```

Passwords / logins

- If you are logged in with any login other than `nmadm`, you will be required to enter a password to execute this command.
- If you are logged in as `nmadm`, no password is required.
- If you have purchased Feature 22, “NMADM Login Accountability” you will not be prompted for the `nmadm` password.

Examples

Use the following command line to activate BDR:

```
$ /nm/sys/bdr_act
```

Related command(s)

“bdr_deact” (p. 10)

□

bdr_audit

Description

The `bdr_audit` (audit BDR) command is used on the host to simulate the execution of the `audit all all` for all collectible entities in backup mode. As part of the normal BDR activities, audit results are transmitted from the primary host to the backup host and are installed in the database. The `bdr_audit` command uses these audit result files to repopulate the database after it has been reconfigured (e.g., after `create all`).

The `audit` command will only run audits for entities in primary mode.

As part of the normal BDR activities, audit results are transmitted from the primary host to the backup host and are installed in the database and also stored as *Linux* files on disk. The `bdr_audit` command populates the database using the *Linux* audit result files that have been backed up from the primary host.

1. The audit information for a backup entity is populated in the database only if the *Linux* audit result files for the entity exist on this host. `bdr_audit` does not access any of the entities or other hosts.
2. The `bdr_audit` command should be executed whenever `audit all all` is run.

Syntax

```
bdr_audit
```

Examples

```
$  bdr_audit
```

Related command(s)

"audit" (p. 16)

□

bdr_chgstat

Description

The bdr_chgstat (change BDR state) command is used on the host to change or show the BDR state of global record base files on an NTM host.

The BDR state of the global file name will not change if:

- It is already in that state.
- The -p option is chosen and another NTM host is already in primary state for this global file and the communication link between the two hosts is up and running.
- The -b option is chosen for a global file in PRIMARY state and the file is flagged because it has not been synchronized with the other hosts (i.e. because the link to another host is down).

If a global file is in PRIMARY state and the file is flagged because it has not been synchronized with the other hosts (i.e. because the link to another host is down), you can remove the flag by changing the state of the global file to INACTIVE.

The command will prompt for arguments if none are provided and will print help messages if “?” is entered.

Syntax

```
bdr_chgstat option filenames
```

Parameters

option Valid values are:

- -b — Changes the state to BACKUP. In the BACKUP state, the file will not be backed up to any other hosts. Another host in PRIMARY state for the same global file will be permitted to back up its changes for the global file to this host.
- -i — Changes the state to INACTIVE. In the INACTIVE state, the file will not be backed up to any other hosts and no other host will be permitted to back up its changes for the global file to this host.
- -p — Changes the state to PRIMARY. In the PRIMARY state, the file will be backed up to all other hosts where this file is in BACKUP state. The contents of the file in other hosts will be overwritten with one from this host. The backup will be performed during end of day procedures or when the create command is executed for the global file. Other hosts will not be permitted to overwrite the content of a global file on this host.
- -s — Displays the current state.

filenames Valid values are:

- name — The full or relative pathname (to “/musr/rb” directory) of the global file. For example, global file “/musr/rb/discrete/ess5” can also be entered as “discrete/ess5”. The pathname is necessary to distinguish global files that have the same name but reside in different directories.
- all — Performs the procedure specified for all global files defined on this host.

Examples

The following command would be used to change the state of the ess5 discrete file to inactive:

```
$ bdr_chgstat -i /musr/rb/discrete/ess5
```

Related command(s)

“create” (p. 5)
“dbtest” (p. 17)

□

bdr_commit

Description

The `bdr_commit` (commit BDR) command is used on the host to synchronize the changes made to global record base files. It backs up the global record base files in PRIMARY state on an NTM host to all other hosts where the same files are in BACKUP state. Global record base files are common for all NTM hosts and therefore can be modified by only one host at a time.

The `bdr_commit` command is especially useful for synchronizing independent files such as the "dcc_alias" file, which cannot be synchronized with the `create` or `dbtest` commands.

Syntax

```
bdr_commit argument
```

Permissions

To execute the `bdr_commit` command, you must have "read" permissions for the specified file on the host where the file is in the PRIMARY state and write permission on the host(s) where the file is in the BACKUP state.

Parameters

`argument` Valid values are:

- all — Performs the procedure specified for all global files defined on this host that are in PRIMARY state.
- global_file_pathname — The full or relative (to "/musr/rb" directory) pathname of the global file. For example, global file "/musr/rb/rspte/typxref" can also be entered as "rspte/typxref." The pathname is necessary to distinguish global files that have the same name but reside in different directories.

Examples

```
$ bdr_commit /musr/rb/rspte/typxref
```

Related command(s)

"bdr_chgstat" (p. 7)

□

bdr_deact

Description

The bdr_deact (deactivate BDR) command is used on the host to display the current status of the BDR (active, not active, or undefined) and also allows you to change the status of BDR to inactive on the NTM host. This will disable the application software on this host from performing backups to other hosts and vice versa. For the host to perform backups to other hosts, BDR must be in the active state.

Syntax

```
/nm/sys/bdr_deact
```

Passwords / logins

- If you are logged in with any login other than nmadm, you will be required to enter a password to execute this command.
- If you are logged in as nmadm, no password is required.
- If you have purchased Feature 22, "NMADM Login Accountability" you will not be prompted for the nmadm password.

Examples

Use the following command line to deactivate BDR:

```
$  /nm/sys/bdr_deact
```

Related command(s)

"bdr_act" (p. 5)

□

bdr_switchbk

Description

The `bdr_switchbk` command is used on the host to switch a host, a Data Collection Concentrator (DCC) and its associated offices, or any direct-connect offices that are in takeover mode back to normal mode. The `bdr_switchbk` command deactivates all the offices associated with the given remote host, the given DCC, or any direct-connect offices accordingly. The remote host can now resume its original function.

Syntax

`bdr_switchbk entity=`*`entity`*

Passwords / logins

You must be logged in as `nmadm` or have `snm` group permissions to execute this command.

Parameters

`entity` The name of the entity you wish to switch back to normal mode. This entity can be the name of a remote host, or a DCC office, that is specified as backup.

System responses

Figure 1 shows an example of the `bdr_switchbk` command output for a DCC.

Figure 1 bdr_switchbk output — DCC

```
$ bdr_switchbk dcc7s
IP

Please remember to run the deact_dcc dcc7s command in order to deactivate
   all the offices known to be associated with this DCC. The execution of
   this command ONLY puts the DCC and its end offices in normal mode.  It
   does not deactivate the individual offices.
$
```

Figure 2 shows an example of the `bdr_switchbk` command output for a host.

Figure 2 bdr_switchbk output — host

```
$ bdr_switchbk hostB
   BDR 308   nms hostA bdr_switchback: Could not get status of NTM
        application on hostB, switchback is permitted
office1 deactivated for audit discrete control meas
office2 deactivated for audit discrete control meas
office3 deactivated for audit discrete control meas
```

```
office4 deactivated for audit discrete control meas
office5 deactivated for audit discrete control meas
office6 deactivated for audit discrete control meas
office7 deactivated for audit discrete control meas
office8 deactivated for audit discrete control meas
Do you wish to enable/disable an exception/line printer(y/n)? n
$
```

Figure 3 shows an example of the `bdr_switchbk` command output for a directly connected office.

Figure 3 bdr_switchbk output — office

```
$ bdr_switchbk sndgca0291t
IP
sndgca0291t deactivated for audit discrete control meas
$
```

Related command(s)

"bdr_takeover" (p. 15)
"linkstat" (p. 9)

□

bdr_sync

Description

The bdr_sync (synchronize BDR) command is used on the primary host to synchronize the record base and audit response files from the backup host to the primary host.

The bdr_sync command is useful when:

- A disaster has destroyed the data on the primary host and no other backups are available.
- The backup host took over the operation of the primary host (because of a disaster) and changes were made to the record base files usually managed by the primary host. The primary host has recovered from disaster and wishes to operate using the modified record base files for the primary host which resides on the backup host.

Syntax

```
/nm/sys/bdr_sync hostname
```

Permissions

Before executing this command, you must have permission to read record base files on the backup host and permission to write record base files on the primary host.

Detailed description

The command will prompt the user for each primary record base partition to verify if it should be synchronized.

This command should run in conjunction with the recreate command to synchronize all record base files on this host. The command will also synchronize all global record base files from the backup host to the primary hosts that are in PRIMARY state.

If the user wants to copy the files from the primary partition on a host to its backup partition on the backup host, this command will not do this function. The user may run create all on the primary host, which will copy the record base files to the backup host.

The synchronization procedure will not be performed if this host is in takeover mode for the specified host.

Important! The bdr_sync command can only be executed from “ksh”. If executed in “sh,” this program will exit.

⚠ **CAUTION**

Because this procedure overwrites the files of the host on which it is executed, it is recommended that you back up the record base first, following the procedures in “Backing up the host with BDR active” (p. 56) in the *System Administration Guide*.

Parameters

hostname	Name of the host with whose record base files you wish to synchronize

Examples

```
$  /nm/sys/bdr_sync  host2
```

Related command(s)

“bdr_chgstat” (p. 7)
“bdr_takeover” (p. 15)
“recreate” (p. 63)

□

bdr_takeover

Description

The `bdr_takeover` command is used on the host to allow this host to take over the function of a failed host, Data Collection Concentrator (DCC), or any direct-connect offices. It identifies the appropriate offices from the failed host and activates them so that data collection may be performed for them on the remote host. This host must be physically set up to take over the offices that exist on the failed host, DCC, or any direct-connect offices.

Syntax

```
bdr_takeover entity=entity
```

Passwords / logins

You must be logged in as `nmadm` or have `snm` group permissions to execute this command.

Before you begin

Before running `bdr_takeover`, ensure that all processes such as record base installdb and audits are not running. This may be done by running the following two *Linux* commands:

```
ps -ef | grep installdb
ps -ef | grep audit
```

> **Important!** If these commands do not detect running processes, `bdr_takeover` may be executed.

To avoid loss of data due to broken links between the take over host and the DCCs, periodically verify that the DCC's can be connected prior to taking over a host either because of a failed or scheduled maintenance takeover.

Parameters

`entity`	The name of the host, a DCC office, or entity you wish to take over.

System responses

Figure 4 shows an example of the `bdr_takeover` command output for a DCC.

Figure 4 bdr_takeover output — DCC office

```
$ bdr_takeover dcc7s
```

```
Please remember to run the act_dcc dcc7s command in order to activate all
   the offices known to be associated with this DCC. The execution of
   this command ONLY puts the DCC and its end offices in takeover mode.
   It does not activate the individual offices.
$
```

Figure 5 shows an example of the `bdr_takeover` command output for a host.

Figure 5 bdr_takeover output — host

```
$ bdr_takeover hostB

   BDR 308   nms hostA bdr_takeover: Could not get status of NTM
   application on hostB, takeover is permitted
office1 activated for audit discrete control meas
office2 activated for audit discrete control meas
office3 activated for audit discrete control meas
office4 activated for audit discrete control meas
office5 activated for audit discrete control meas
office6 activated for audit discrete control meas
Do you wish to enable/disable an exception/line printer(y/n)? n
```

Figure 6 shows an example of the `bdr_takeover` command output for a directly connected office.

Figure 6 bdr_takeover output — office

```
$ bdr_takeover
entity = sndgca0291t
  IP
sndgca0291t activated for audit discrete control meas
$
```

Related command(s)

"bdr_switchbk" (p. 11)
"linkstat" (p. 9)

□

sync_audit

Description

The `sync_audit` (synchronize audit controls) command synchronizes the data base controls that reside in the secondary subnetwork partitions on this NTM host.

The `sync_audit` command is useful when host “A” has taken over offices from host “B” and controls are added to offices from host “B”. After host “B” takes back the offices from host “A”, the user deletes the controls in the offices and the host “B” database. However, the controls remain in the host “A” database and audits on host “A” cannot remove the controls because the offices are not on host “A”. As a result, database controls on hosts “A” and “B” are out of sync.

This command should be run to synchronize all database controls on the secondary partitions on this host with the other hosts.

The synchronization procedure will not be performed if this host does not have the BDR feature.

Syntax

```
/nm/cmdbin/sync_audit
```

Restrictions

The `sync_audit` command can only be executed from “ksh”. If executed in “sh,” this program will exit.

Permissions

Before executing this command, you must have permission to read audit response files on the secondary subnetwork partitions on the same host.

CAUTION

Because this procedure overwrites the data base controls of the host on which it is executed, it is recommended that it be run only

if controls remain in the data base for offices that are in the secondary partitions.

Related Command(s)

“bdr_audit” (p. 6)
“bdr_takeover” (p. 15)

□

4 Control Commands

Overview

Purpose

This chapter lists and describes the NTM control commands, as follows:

- Control commands — commands that affect traffic flow. These are: `acc`, `canf/cant/skip`, `cg`, `control`, `cr`, `cro`, `dptpri`, `dptres`, `dpttid`, `rc`, `rr`, `silc`
- Control-related commands — commands that display or modify information about the control commands or provide additional data for support in applying controls. These are: `ctrlactlist`, `ctrlog`, `dhtr`, `ihtr`, `ofcovrd`, `pp`, `pplist`, `preplan`, `purglog`, `total`

These commands have been arranged in this way in order to eliminate duplicate descriptive information.

> **Reference:** See Chapter 6, "Audits and Controls" in the *System Overview* for overview information on controls and how they are used in NTM.

Contents

This chapter contains the following topics:

□

Control commands

Overview

Purpose

Controls are messages exchanged between NTM and the monitored network offices to manage traffic congestion on the network.

Contents

This section contains the following topics:

□

Background

Control types

Controls are manual or automatic.

Manual controls

Manual controls are placed on objects such as trunk groups or destinations.

- A trunk group control is either protective or expansive
  - Protective — restricts normal trunk group access by eliminating or reducing routing alternatives to prevent traffic from reaching congested areas.
  - Expansive — permits calls to be completed over additional non-standard routing patterns to bypass congested trunk groups and/or switching systems. Reroute is the only expansive control available to the network manager.
- A destination (code) control restricts the number of calls that will be allowed to complete to a specified code.

Automatic controls

Automatic controls are manually assigned to objects such as trunk groups or destinations.

The controls do not activate, however, until certain predetermined conditions are met. Then they will automatically initiate control action. The automatic controls will also automatically cease control action when the conditions improve.

Why use controls?

Depending on the control type, it is possible to control traffic on combinations of traffic destination and traffic characteristics (HTR/NHR, AR/DR, etc.). This greatly improves the efficiency of traffic management by allowing the network manager to selectively control traffic during overload such that higher priority calls and/or calls with the best completion probabilities receive favorable treatment.

How are controls implemented?

Network managers use control pages, control commands at the command line, or control preplans to implement controls.

NTM maintains an online control log for recording active and matched (deleted) controls.

□

Using parameters with control commands

Overview

If you enter a control command without any parameters, the system prompts you for all required parameter values. The required parameters are determined by the control action (`act` parameter) and the switch type and generic.

Whether or not a parameter is required may also depend on the other parameters you choose. Mutually exclusive or dependent parameters are noted in the Parameters sections for each command.

Parameter requirements

The following list describes the parameter requirements for adding, modifying, and deleting a control:

- `act=add`

  All parameters are required.
- `act=del`

  Only those parameters that are not shown as being optional (shown in square brackets) in the command message format are required. These are the key parameters of the control.
- `act=mod`

  Only those parameters that are not shown as being optional (shown in square brackets) in the command message format are required. These are the key parameters of the control. However, the command prompts you for all values to allow modifications. Press **RETURN** to indicate no modification and enter a semicolon (`;`) to stop the prompting.

  Important! Key parameters cannot be modified.

Prompting

The following applies to the prompting of parameter values for each command:

- Entering **RETURN** defaults a parameter.
- Entering a semicolon (**;**) indicates you have entered your last parameter value.
- Any optional parameters not specified use their specific default values.

  Important! If a parameter cannot be defaulted, you are prompted to enter an acceptable value.

Execution mode

The last parameter (mode) of the control commands specifies an execution mode. You are only prompted for this parameter if you specify the exclamation point (!) as your terminating character (because mode is optional). However, if you want to validate the control, you can specify this parameter on the command line with one of the following values:

- exec (execute the control; this is the default — it specifies that command processing take place upon command entry)
- validate (validate the control; this verifies the syntax of the parameters entered but does not process the command)

Examples

For example, if you want to verify a set of parameters for use with the cg command without activating a Manual Call Gapping Control in the specified switch, you can use the validate mode.

In Figure 1, the cg command syntax specifies validate for the mode parameter. This will validate the command syntax without activating a Manual Call Gapping Control in the switch.

Figure 1 Command syntax in validate mode

```
$ cg office=clmb04 act=mod code=459 gap=50 type=pots tg=none
   mode=validate
```

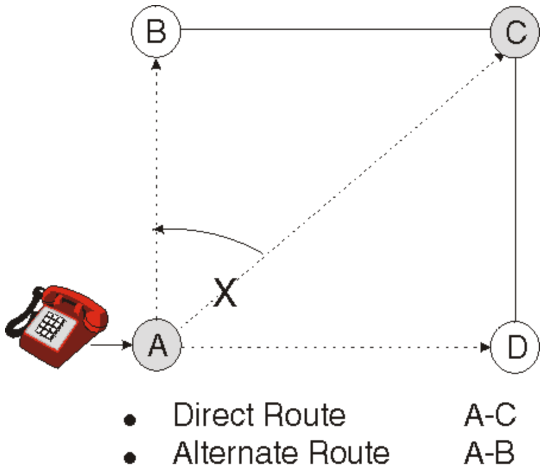
□

Pre-hunt / post-hunt controls

Pre-hunt

A pre-hunt control may affect a call ***before*** it is offered to a trunk group. In Figure 2, the calls are blocked before they are offered to the A-C trunk group.

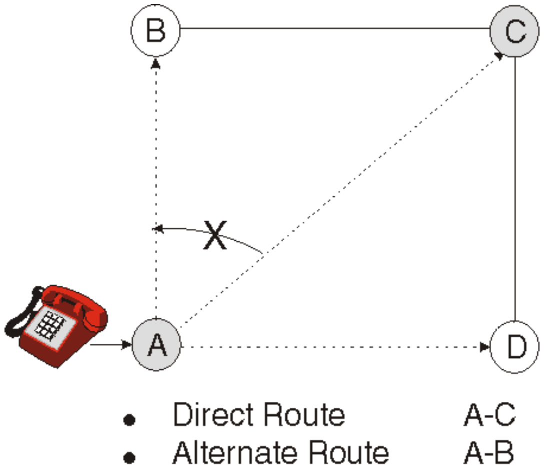
Figure 2 Pre-hunt controls



Post-hunt

A post-hunt control may affect a call that is attempting to alternate route to the next designated trunk group (overflow). In Figure 3, the call is blocked from alternate routing on the A-B trunk group. Blocked calls are sent to an announcement. Based on switch type, calls may be blocked on a percentage basis.

Figure 3 Post-hunt controls



□

List of control commands

Table

Table 1 lists each NTM control command and its function.

Table 1 Control commands and functions

Command	Function	Office Restrictions
`acc`	Applies or removes a “automatic congestion control” command	*4ESS*, *5ESS*, 7R/E
`canf` (`canf/cant/skip`)	Applies, modifies, or removes a “cancel from” control	
`cant` (`canf/cant/skip`)	Applies, modifies, or removes a “cancel to” control	
`cg`	Applies, modifies, or removes a “call gap” control	
`control`	Modifies controls for nextonersm	*NexTone*
`cr`	Applies, modifies, or deletes a “circuit reservation” control	1A *ESS, 4ESS, 5ESS*, 7R/E, *GTD-5,* Sonus
`cro`	Applies or removes a “cancel reroute overflow” control	*4ESS, 5ESS*, 7R/E
`dptpri`	Activates, modifies, or removes a “dynamic packet trunks prioritization” control	Succession
`dptres`	Activates, modifies, or removes a “dynamic packet trunks reservation” control	Succession
`dpttid`	Activates, modifies, or removes a “dynamic packet trunks terminal identifiers” control	Succession
`rc`	Applies routing code for call gapping	*GTD-5*
`rr`	Applies, modifies, or deletes a “reroute” control	
`silc`	Enables or disables a selective incoming overload control	1A *ESS, 5ESS*, 7R/E, *GTD-5*
`skip` (`canf/cant/skip`)	Applies, modifies, or deletes a “skip” control	

□

acc

Description

The `acc` (automatic congestion control) command establishes, inhibits, restores, modifies, or removes the control parameters on an outgoing or a 2-way trunk group out of a particular “From Office”.

The ACC control is an automatic control that senses congestion at a toll or tandem office and activates a Pre-Hunt control (Cancel, Skip, or Reroute) in offices sending traffic to the affected switch. Affected calls receive the activated controls normal response. The ACC control goes into effect when the switch processor enters either the MC1 (machine congestion level 1) or MC2 (machine congestion level 2) level of processor overload. When the overload is reduced below MC1 level, the control is removed.

Syntax

```
acc from=from act=action tg=office-suffix ih/rs=inhibit/restore
   [sk/cn=sk/cn] [cat=category] [mode=mode]
```

Switch restrictions

This command applies only to *4ESS*, *5ESS*, and 7R/E switch types.

Parameters

`from` “From Office” name

`action` Required action to be taken. Valid values are:

- add = add an ACC control to a trunk group in the “From Office”
- mod = modify an ACC control on a trunk group in the “From Office”
- del = delete an ACC control from a trunk group in the “From Office”

`office-suffix` “To Office” name plus a suffix (up to 4 characters) used to identify the trunk group the control is associated with; separate the office name from the suffix with a hyphen (-).

`inhibit/restore` Valid values are:

- in = inhibit
- rs = restore (default)

`sk/cn` Valid values are:

- sk = skip (default)
- cn = cancel

`category` Response category.

- For *4ESS* switches, valid values are: a, b, c, d, e, f, g, h, i, j, k, l, m, n.

  Reference: Table 2

- For *5ESS* and 7R/E switches, valid values are: a, b, c, d, e, f.

  Reference: Table 3

`mode` Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Tables

The following tables provide more information about the parameters for the `acc` command.

Table 2 4ESS control response categories for ACC and CR (Sheet 1 of 2)

CR L1 and ACC MC1 Responses:				
Category	**DR-NHR**	**AR-NHR**	**DR-HTR**	**AR-HTR**
A	0	0	0	100
B	0	0	0	100
C	0	0	75	100
D	0	100	0	100
E	0	0	75	100
F	0	0	0	0
G	0	0	0	0
H	0	0	0	100
I	0	0	100	100
J	0	0	100	100
K	0	100	100	100
L	50	75	50	75
M	0	0	0	0
N	0	0	0	0

Table 2 4ESS control response categories for ACC and CR (Sheet 2 of 2)

CR L2 and ACC MC2 Responses:				
Category	**DR-NHR**	**AR-NHR**	**DR-HTR**	**AR-HTR**
A	0	0	75	100
B	0	100	0	100
C	0	100	75	100
D	100	100	100	100
E	0	0	87	100
F	0	0	0	100
G	0	0	100	100
H	0	0	100	100
I	0	100	100	100
J	75	100	100	100
K	75	100	100	100
L	75	75	75	75
M	0	0	0	0
N	0	0	0	0

Notes: Definitions for control response categories are as follows:

- AR — Alternate Routed
- DR — Direct Routed
- HTR — Hard-To-Reach
- NHR — Non HTR

Table 3 5ESS and 7R/E ACC control response percentages

ACC Congestion Level	Traffic Types	Response Category					
		A	**B**	**C**	**D**	**E**	**F**
CL1	AR_NHR	0	0	75	0	0	100
	DR-NHR	0	0	50	0	0	0
	AR-HTR	0	0	100	100	100	100
	DR-HTR	0	0	0	0	100	100

Table 3 5ESS and 7R/E ACC control response percentages

ACC Congestion Level	Traffic Types	Response Category					
		A	B	C	D	E	F
CL2	AR_NHR	0	0	75	100	100	100
	DR-NHR	0	0	75	0	75	75
	AR-HTR	100	100	100	100	100	100
	DR-HTR	0	100	100	100	100	100

Notes: Definitions for control response categories are as follows:

- AR — Alternate Routed
- DR — Direct Routed
- HTR — Hard-To-Reach
- NHR — Non HTR

There is an additional response category for the 7R/E. However, this cannot be implemented through the GUI.

Examples

```
$ acc from=colsoh5040 act=add tg=colsoh4e003 010 ih/rs=rs
  sk/cn=sk cat=a
```

“ctrlog” (p. 59)

□

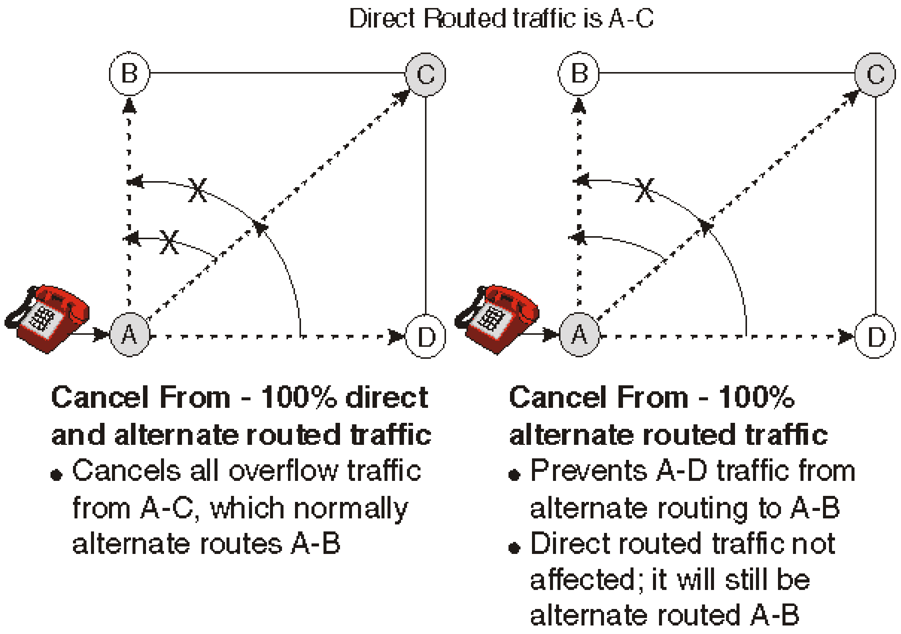
canf/cant/skip

Description

canf

The `canf` (cancel from) command activates, modifies, or removes a CANF control on an outgoing or a 2-way trunk group out of a particular “From Office”. A CANF control is a Post-Hunt control that inhibits a call from hunting for an idle trunk in subsequent trunk groups in the routing chain when an overflow condition occurs. If no idle circuit is found, the call is sent to an announcement.

Figure 4 Graphical representation of CANF control



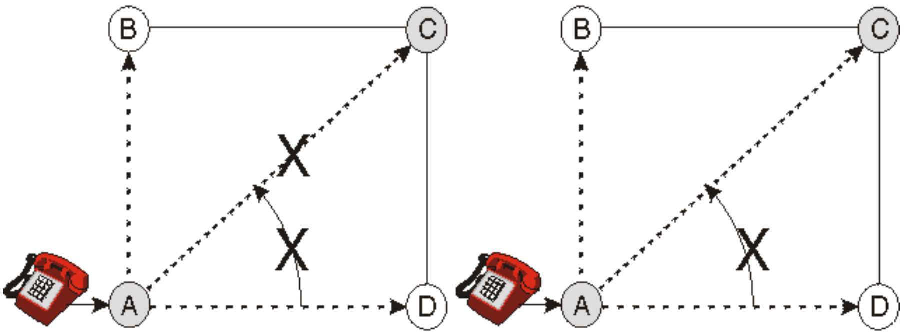
Direct Routed traffic is A-C
B
C
A
D
X
X
B
C
A
D
X
Cancel From - 100% direct and alternate routed traffic
• Cancels all overflow traffic from A-C, which normally alternate routes A-B
Cancel From - 100% alternate routed traffic
• Prevents A-D traffic from alternate routing to A-B
• Direct routed traffic not affected; it will still be alternate routed A-B

cant

The `cant` (cancel to) command activates, modifies, or removes a CANT control on an outgoing or a 2-way trunk group out of a particular “From Office”. A CANT control is a Pre-Hunt control that prevents a trunk group from being searched for idle trunks and routes affected traffic to an announcement.

Figure 5 Graphical representation of CANT control

Direct Routed traffic is A-C



Cancel To - 100% direct and alternate routed traffic

- Cancels A-D overflow traffic
- Cancels A-C direct routed traffic

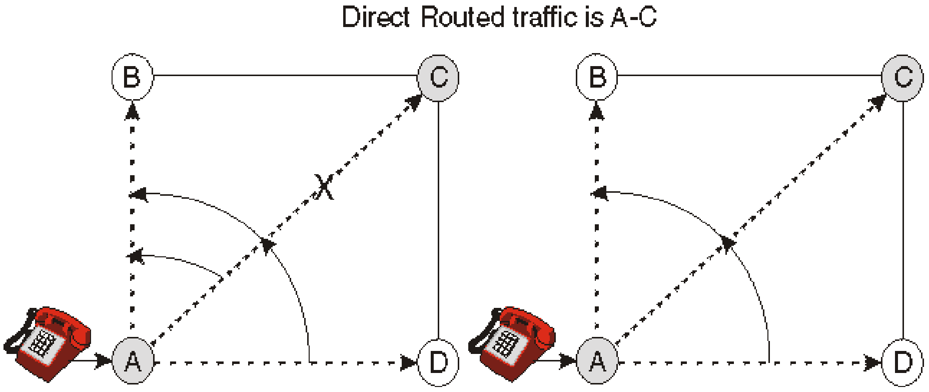
Cancel To - 100% alternate routed traffic

- Prevents A-D traffic from alternate routing to A-C
- Direct routed traffic not affected

skip

The `skip` command activates, modifies, or removes a SKIP control on an outgoing or a 2-way trunk group out of a particular “From Office”. A SKIP control is a Pre-Hunt control that inhibits a trunk group from being searched for an idle trunk by passing the call to the next normal group in the routing chain or table. It can also be used to pass a call to an out-of-chain group from the last normal in-chain group.

Figure 6 Graphical representation of SKIP control



Skip - 100% direct and alternate routed traffic

- Direct routed traffic will skip A-C and go to A-B
- Overflow traffic from A-D will skip A-C and go to A-B

Skip - 100% alternate routed traffic

- Overflow traffic from A-D will skip A-C and go to A-B
- Direct routed traffic is not affected

Syntax

Since the input parameters vary according to the switch and generic, the command syntax is listed for each.

Important! The annc= parameter is not valid for the skip command.

Switch Type(s)	Generic(s)	Syntax
1A *ESS*		*COMMAND* from=*from* act=*action* tg=*office-suffix* [alt=*alt*] [dir=*dir*] [annc=*announcement*] [mode=*mode*]
5ESS	5e4	
DMS 100/200		
DMS 250		
DMS 500		
SCSN	sn02	
LSSGR		
EWSD		
Sonus	gsx5_1+	
GSP	gsp07+	

Switch Type(s)	Generic(s)	Syntax

4ESS	4e12	`COMMAND from=from act=action tg=office-suffix`
		`[alt=alt] [dir=dir] [htr=htr]`
		`[annc=announcement] [mode=mode]`

4ESS	4e13+	`COMMAND from=from act=action tg=office-suffix`
		`[tfc=tfc] [rout=rout] [alt=alt] [dir=dir]`
		`[htr=htr] [annc=announcement] [mode=mode]`

5ESS	5e5+	`COMMAND from=from act=action tg=office-suffix`
		`[alt=alt] [dir=dir] [annc=announcement]`
		`[i/x=inc/xcl] [domains=domains] [mode=mode]`

7R/E		

SCSN	sn03+	`COMMAND from=from act=action tg=office-suffix`
		`[halt=halt] [hdir=hdir] [ealt=ealt] [edir=edir]`
		`[annc=announcement] [mode=mode]`

GTD-5	gtd1641	

GTD-5	gtd1711 +	`COMMAND from=from act=action tg=office-suffix`
		`[rout=rout] [halt=halt] [hdir=hdir] [ealt=ealt]`
		`[edir=edir] [annc=announcement] [mode=mode]`

Parameters

`COMMAND` Commands are:

- `canf` = Cancel From
- `cant` = Cancel To
- `skip` = Skip

`from` "From Office" name

`action` One of the following required actions:

- add = add a `COMMAND` control to a trunk group in the "From Office"
- mod = modify a `COMMAND` control on a trunk group in the "From Office"
- del = delete a `COMMAND` control from a trunk group in the "From Office"

`office-suffix` "To Office" name plus a suffix (up to 4 characters) used to identify the trunk group with which the control is associated; separate the office name from the suffix with a hyphen (-).

`tfc` One of the following required actions:

- alt = alternate routed traffic
- dar = direct and alternate routed traffic
- dir = direct routed traffic (4e17 and later only)

Reference: Note 1

`rout` Control method for selecting controlled calls. Allowable values are:

- pct = percent of calls to control (default)
- rate = maximum rate of calls to control in the 5-minute interval (4e13 and later, *GTD-5* 1711 and later only)

  Reference: Note 1

`alt` Percent or rate of alternate routed traffic to control.

Reference: Note 1, Note 2, Table 4

`dir` Percent or rate of direct routed traffic to control.

Reference: Note 2, Table 4

`htr` Hard-to-reach option (*4ESS* only):

- htr = Hard-to-reach
- h-u = High usage

`halt` Percent or rate of hard-to-reach alternate routed traffic to control.

Reference: Note 3, Table 4

`hdir` Percent or rate of hard-to-reach direct routed traffic to control.

Reference: Note 3, Table 4

`ealt` Percent or rate of easy-to-reach alternate routed traffic to control.

Reference: Note 3, Table 4

`edir` Percent or rate of easy-to-reach direct routed traffic to control.

Reference: Note 3, Table 4

`announcement` Announcement type (the default is `nca`).

Important! This parameter does not apply to the `skip` command.

Reference: Table 4

`inc/xcl` Include and exclude domains (*5ESS* generic 5e5 and later only)

- inc = Include (default)
- xcl = Exclude

`domains` List of up to 5 domains or all (default) (*5ESS* generic 5e5 and later only)

Important! This value is applicable only when `i/x=inc`.

Reference: These values are defined in the Record Base. See “Domain Acronym File” (p. 19) and “Office Domain File” (p. 54) in the *Record Base Administration Guide*.

`mode` Execution mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Notes

1. When rout=pct, the tfc and alt parameters are linked, and both must be specified.
2. The alt and dir parameters are linked keywords.
   - For all switch types except *4ESS* and *GTD-5* switches, you must specify alt, dir, or both, for the control to function. The values entered can be different.
   - For *4ESS* switches prior to 4e17, the dir parameter cannot be specified alone. When you desire to control all traffic, enter the same value for both the alt and dir parameters. If a different value is entered, the value in the alt parameter will be used as the value to control both alternate and direct routed traffic.
   - In 4e13 and later switches, the alt and dir parameters are linked to the tfc parameter.
   - For 4e17 and later switches, specifying tfc=dir will allow the dir parameter to be specified alone. If tfc=dar is selected, you must enter a value in both the alt and dir parameters, but, as in previous generics, the alt value will be used for the control and the dir value will be ignored.
3. At least one of the halt, hdir, ealt and edir parameters must have a non-zero value for the control to function.

Table

Table 4 lists the valid parameter values for parameters whose values differ by switch type.

Table 4 CANF/CANT/SKIP control table

Variable	Generic or Switch	Values
alt dir	1A *ESS*	0, 50, 75, 100
	4ESS	25, 50, 75, 87, 100
	4ESS if rout=rate	.5, 1, 1.5, 2, 2.5, 3, 4, 5, 6, 8, 10, 15, 30, 45, 60
	5e14 and earlier, *DMS, EWSD,* LSSGR, SCSN, *Sonus* gsx5_1 and later	0, 12, 25, 37, 50, 62, 75, 87, 100
	5e15 and later, 7R/E, GSP	0–100, inclusive, in increments of 1
halt hdir ealt edir	*GTD-5* 1641 *GTD-5* 1711 if rout=pct	0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100
	GTD-5 1711 if rout=rate	0–60, 120, 300, 600, p5, inf
	DMS, SCSN	0, 12, 25, 37, 50, 62, 75, 87, 100
	7R/E ivtoa 4_1 and 5e16_1h	0–100, inclusive, in increments of 1

Table 4 CANF/CANT/SKIP control table (continued)

Variable	Generic or Switch	Values
`announcement` (does not apply to the `skip` command)	1A *ESS*	nca
	4e12, 4e13	ea1, ea2, ea3, ea4, ea5, ea6, ssb, nca, vca, inc
	4e14 through 4e19	ea1, ea2, ea3, ea4, ea5, ea6, ea7, ssb, nca, vca, inc, roa, sp1, sp2, mca, ncc, nwc, oob, ncd, ivc, iff
	4e20 and later	ea1, ea2, ea3, ea4, ea5, ea6, ea7, ssb, nca, vca, inc, roa, sp1, sp2, mca, ncc, nwc, oob, ivc, iff, sna, vma, nwn, dnn, tdn, inf, wat, lsa, ooa, lob, unn, doo, e01, e02, e03, e04, e05, e06, e07, e08, e09, e10, e11, e12, e13, e14, e15, e16, e17, e18, e19, e20, r01, r02, r03, r04, r05, r06, r07, r08, r09, r10, i01, i02, i03, i04, i05, i06, i07, i08, i09, i10, i11, i12, i13, i14, i15, i16, i17, i18, i19, i20, i21, i22, i23, i24, i25, i26, i27, i28, i29, i30
	5ESS, 7R/E, *DMS, EWSD,* LSSGR, SCSN, *Sonus* gsx5_1 and later, GSP	ea1, ea2, nca
	GTD-5	nca, roa, rot, fea, sea

Examples

The following is an example of a `canf` command for a *DMS*, *EWSD*, LSSGR, or 1A *ESS* switch:

```
$ canf from=colsohdms24 act=add tg=clevoh5es10-001 alt=50
   dir=50 annc=nca
```

The following is an example of a `skip` command for a 4e13 alternate routed control:

```
$ skip from=colsoh4es04 act=add tg=colsohdms24-017 tfc=alt
   rout=pct alt=25 htr=h-u
```

The following is an example of a `cant` command for a 5e5 alternate and direct routed control:

```
$ cant from=clevoh5es10 act=add tg=colsohdms24-003 alt=50
   dir=25 annc=nca i/x=inc domains=pots
```

The following is an example of a `canf` command for a *GTD-5* switch:

```
$ canf from=dllstxg1594 act=add tg=ftwttxg0384-001 halt=50
   hdir=25 ealt=50 edir=25 annc=nca
```

Related command(s)

“ctrlog” (p. 59)
“total” (p. 90)

□

Description

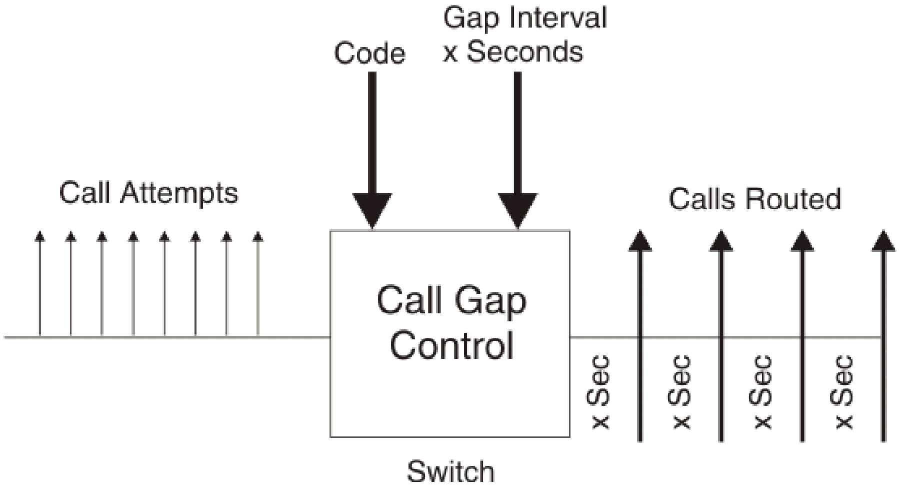
The `cg` (call gap) command applies a CG control to:

- All calls with routing digits matching the specified “controlled digits” in a particular office(s)
- All calls with routing digits matching the specified “controlled digits” received from a specified incoming TSG (Trunk Subgroup) in a specific *4ESS* switch

The CG control is a Pre-Hunt code control that regulates the number of calls allowed to proceed normally to a specific telephone number or code. Affected calls are sent to an announcement.

Depending on switch type, calls may also be gapped on the basis of announcement type, feature group, prefix digit type, and domain or type of network traffic.

Figure 7 Graphical representation of CG control



Syntax

Since the input parameters vary according to the switch and generic, the command syntax is listed for each.

Important! You must supply an `ic`, a `code`, or values for both when these parameters appear in the syntax.

Switch Type(s)	Generic(s)	Syntax

1A *ESS*		`cg office=office act=action [ic=ic] [code=code]`
		`[gap=gap] [annc=announcement] [mode=mode]`

5ESS	5e4	
DMS 100/200		
DMS 250		
DMS 500		
EWSD	ewsd13a +	
LSSGR		
SCSN		
Sonus	psx5_1+	

4ESS	4e12	`cg office=office act=action code=code [gap=gap]`
		`[dom=domain] [annc=announcement] [mode=mode]`

4ESS	4e13 +	`cg office=office act=action code=code [gap=gap]`
		`[type=type] [trap=security trap]`
		`[annc=announcement] [tg=office-suffix]`
		`[mode=mode]`

5ESS	5e5 +	`cg office=office act=action [ic=ic] [code=code]`
		`[gap=gap] [dom=domain] [annc=announcement]`
		`[mode=mode]`

7R/E		

EWSD	ewsd10-13	`cg office=office act=action [code=code]`
		`[gap=gap] [annc=announcement] [mode=mode]`

GTD-5	gtd1641	`cg office=office act=action [pfxt=pfxt] [ic=ic]`
		`[code=code] [ccd=ccd] [pccd=pccd] [gap=gap]`
		`[annc=announcement] ftype=ftype [mode=mode]`

GTD-5	gtd1711 +	`cg office=office act=action cg_typ=cg_typ`
		`[pfxt=pfxt] [ic=ic] [code=code] [ccd=ccd]`
		`[pccd=pccd] [gap=gap] [lgap=lgap] [tgap=tgap]`
		`[annc=announcement] [ftype=ftype] [mode=mode]`

Parameters

`office` Office name

Reference: Note 1

`action` Required action. Specify one of the following:

- add = add CG control to an office
- mod = modify CG control presently on an office

- del = delete CG control from an office

`cg_typ` Call gap type (*GTD-5* 1711 and later only). Specify one of the following:

- dest = destination
- term = terminating

`pfxt` Prefix digit types (*GTD-5* only).

- If `cg_typ=dest`, valid values are: any, iddd, npfx, ie3d, ddd, p950, eddd
- If `cg_typ=term`, valid values are: any, npfx, ddd, eddd

`ic` A 4-digit carrier prefix code. Valid values are:

- 0000–0999 — 1ae11 and earlier, 5e8 and earlier, LSSGR
- 0000–9999 — 1ae12 and later, 5e9 and later, ewsd13a and later, *DMS*, *GTD-5*, Sonus psx5_1 and later

Notes:

- This is optional if the `code` parameter has a value specified.
- *4ESS* switches do not have IC control capabilities through NTM.

`code` Routing number(s) of the call you want to gap.

Important! This is optional if the `ic` parameter has a value specified.

Reference: Note 1, Table 5, “Call gap control table” (p. 24)

`ccd` Country code digits (*GTD-5* only); 1–3 digits in the range 0–999.

`pccd` Pseudo-country code digits (*GTD-5* only); 3-digit code in the range 000–999

`gap` Gap rate (the maximum number of calls allowed through in a 5-minute period) (applies when `cg_typ=dest`).

Reference: Table 5, “Call gap control table” (p. 24)

`lgap` Line-to-line gap rate (applies when `cg_typ=term`).

Reference: Table 5, “Call gap control table” (p. 24)

`tgap` Trunk-to-line gap rate (applies when `cg_typ=term`).

Reference: Table 5, “Call gap control table” (p. 24)

`announcement` Announcement type.

Reference: Table 5, “Call gap control table” (p. 24)

`domain` Domain type (4e12 and *5ESS* switches only)

Reference: These values are defined in the Record Base. See "TYPXREF File" (p. 105) and "Trunk Group Threshold File" (p. 98) in the *Record Base Administration Guide*.

`type` Type of traffic to control. (4e13 and later). Valid values are: dom, pots, npot, sdn, acnt, orig, tran, term, inc, sspd, sspi

`ftype` Valid values are: b, d, c (combined). (*GTD-5* only)

`office-suffix` Incoming trunk subgroup and suffix (*4ESS* only)

`security trap` Call security trap (*4ESS* only). Valid values are: on, off

`mode` Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Notes

1. Enter multiple values separated by the plus sign (+).

Table

Table 5 lists the valid parameter values for parameters whose values differ by switch type.

Table 6 lists digit combinations for the *GTD-5* switch.

Table 5 Call gap control table (Sheet 1 of 2)

Variable	Generic or Switch	Values
`code`	1A *ESS*	Up to 6 digits
	4e21 and earlier	Up to 12 digits
	4e22 and later, 5e9 and earlier, *DMS*, *EWSD*, LSSGR, SCSN, Sonus psx5_1	Up to 15 digits
	5e10 and later, 7R/E	Up to 18 digits
	GTD-5 1641	3 or 6 digits
	GTD-5 1711 and later	10 digits

Table 5 Call gap control table (Sheet 2 of 2)

Variable	Generic or Switch	Values
`gap`	1A *ESS*	all, 3000, 1200, 600, 300, 150, 60, 30, 20, 10, 5, 2.5, 1, .5, none
	4ESS	all, 3000, 2500, 2143, 1875, 1667, 1500, 1200, 833, 600, 500, 400, 300, 188, 100, 50, 33, 25, 20, 15, 10, 7, 5, 4, 3, 2.5, 2, 1, .6, .5, none
	5e5 and earlier	all, 3000, 1200, 600, 300, 150, 60, 30, 20, 10, 5, 2.5, 1, .5, none
	5e6 and later, 7R/E	all, 3000, 2500, 2143, 1875, 1667, 1500, 1200, 833, 600, 500, 400, 300, 188, 150, 100, 60, 50, 33, 30, 25, 20, 15, 10, 7, 5, 4, 3, 2.5, 2, 1, .6, .5, none
	DMS, EWSD, LSSGR, SCSN, Sonus psx5_1	all, 3000, 1200, 600, 300, 150, 60, 30, 20, 10, 5, 2.5, 1, .5, none
	GTD-5 1641	600, 300, 150, 60, 30, 20, 10, 5, 2.5, 1, .5, none
	GTD-5 1711 and later	600, 300, 150, 100, 75, 60, 50, 42.9, 37.5, 33.3, 30, 27.3, 25, 23.1, 21.4, 20, 18.8, 17.7, 16.7, 15.8, 15, 14.3, 13.6, 13, 12.5, 12, 11.5, 11.1, 10.7, 10.3, 10, 9.7, 9.4, 9.1, 8.8, 8.6, 8.3, 8.1, 7.9, 7.7, 7.5, 7.3, 7.1, 7, 6.8, 6.7, 6.5, 6.4, 6.3, 6.1, 6, 5.9, 5.8, 5.7, 5.6, 5.5, 5.4, 5.3, 5.2, 5.1, 5, 2.5, 1, .5, none
`announcement`	4e13 and earlier	ea1, ea2, ea3, ea4, ea5, ea6, ssb, nca, vca, inc
	4e14 and later	ea1, ea2, ea3, ea4, ea5, ea6, ea7, ssb, nca, vca, inc, roa, sp1, sp2, mca, ncc, nwc, oob, ncd, ivc, iff
	1A *ESS*, *5ESS,* 7R/E, *DMS, EWSD,* LSSGR, SCSN, Sonus psx5_1	ea1, ea2, nca
	GTD-5 1641	nca, roa, rot, fea, sea
	GTD-5 1711 and later	nca, roa, rot, fea, sea, busy (busy only applies when `cg_typ=dest`)
`lgap` `tgap`	gtd1711 and later (for terminating call gaps)	600, 300, 150, 100, 75, 60, 50, 42.9, 37.5, 33.3, 30, 27.3, 25, 23.1, 21.4, 20, 18.8, 17.7, 16.7, 15.8, 15, 14.3, 13.6, 13, 12.5, 12, 11.5, 11.1, 10.7, 10.3, 10, 9.7, 9.4, 9.1, 8.8, 8.6, 8.3, 8.1, 7.9, 7.7, 7.5, 7.3, 7.1, 7, 6.8, 6.7, 6.5, 6.4, 6.3, 6.1, 6, 5.9, 5.8, 5.7, 5.6, 5.5, 5.4, 5.3, 5.2, 5.1, 5, 2.5, 1, .5, none

Table 6 Digit combinations for GTD-5 offices

Reference Number	Digit Combinations				PREFIX TYPES						
					ANY	NPFX	DDD	EDDD	IDDD	IE3D	P950
1	PFXT=pfxx	NPA=xxx			X	X	X	X			
2	PFXT=pfxx	NPA=xxx	CO=yyy		X	X	X	X			
3	PFXT=pfxx	NPA=xxx	CO=yyy	SD=xxxx	X	X	X	X			
4	PFXT=pfxx								X	X	
5	PFXT=pfxx	CCD=xxx							X	X	
6	PFXT=pfxx	PCCD=xxx							X	X	
7	IC=xxxx	PFXT=pfxx			X	X	X	X	X	X	X
8	IC=xxxx	PFXT=pfxx	NPA=xxx		X	X	X	X			
9	IC=xxxx	PFXT=pfxx	NPA=xxx	CO=yyy	X	X	X	X			
10	IC=xxxx	PFXT=pfxx	CCD=xxx		X				X	X	
11	RC=xxx	IC=xxxx									
12	RC=xxx	IC=xxxx	CCD=xxx								
13	RC=xxx	ISRC=xx	IC=xxxx								
14	RC=xxx	ISRC=xx	IC=xxxx	CCD=xxx							
15	ISRC=xxx	IC=xxxx									
16	ISRC=xxx	IC=xxxx	CCD=xxx								

Examples

The following is an example of the `cg` command for a *DMS* or a 1A *ESS* switch:

```
$ cg office=dms24 act=add ic=0123 code=614 gap=30 annc=nca
```

The following is an example of the `cg` command for a 4E13 switch:

```
$ cg office=sm01 act=add code=7220735 gap=20 annc=ea2
  type=pots tg=none trap=off
```

The following is an example of the `cg` command for a *5ESS* switch with an “ic” prefix:

```
$ cg office=d5e6 act=add ic=0312 code=6148601212 gap=20
  annc=ea1 dom=pots
```

The following is an example of the `cg` command for a *GTD-5* switch:

```
$ cg office=gtd51711 act=add cg_typ=dest pfxt=any ic=0113
  code=6148601212 gap=2.5 annc=rot ftype=d
```

Related command(s)

“ctrlog” (p. 59)
“total” (p. 90)

□

control

Description

The `control` command modifies controls on User-Defined Network Elements of the switchtype “nextonersm”. The command supports the following actions: adding, modifying, and deleting.

Syntax

```
control type=type office=element act=action [paramName=paramValue]
   [mode=mode]
```

usage for ratelimit control:

```
control type=ratelimit office=element act=action
   [limtype=limittype] [devname=devicename] [part=partition]
   [name=name] [mod=mod] [layer=layer] [limrate=limitrate]
   [limburst=limitburst] [ulog=ulograte] [thresh=threshold]
   [mode=mode]
```

usage for ratelimitpolicy control:

```
control type=ratelimitpolicy office=element act=action
   devname=devname policy=policy
   [outcallthreshold=outcallthreshold]
   [outcallinterval=outcallinterval] [mode=mode]
```

usage for OCL controls:

```
control type=ocl[dtn|endpnt|realm] office=element act=action
   devname=devname
   dtn|endpnt|realm=someDTN|someEndPoint|someRealm
   [rateLimitPolicy=rateLimitPolicy] [mode=mode]
```

Switch restrictions

This command is only valid for *nextone* switch type.

Parameters

`type` Control type.

`element` Network Element name.

> **Important!** In case of ratelimitpolicy, a value of all will indicate that all RSMs should be addressed.

`action` Required action. For ratelimit control only action “mod” is allowed. Specify one of the following:

- add - add control to an office

  Important! In case of ratelimitpolicy, if act is set to add, then outcallthreshold and outcallinterval are required parameters. If act is del, outcallthreshold and outcallinterval are ignored if present. If act is set to mod, then one of those two parameters must be set. If act is del, the command first verifies that the rateLimitPolicy is not currently in use by an active control before attempting the del.

- mod – modify control presently on an office.
- del – delete control from an office

`devname` devname is the name of the S/BC. A value of all will indicate that the message should be sent for each S/BC.

`dtn` Destination telephone number. It is a string from an NTM point of view.

`endpnt` End Point. It must be a valid end point andpoint for this S/BC.

`realm` Realm. It must be a valid realm for this S/BC.

`policy` Name of the rateLimitPolicy.

`rateLimitPolicy` The name of the rateLimitPolicy (or null to turn the control off). The value must be from the set of active rateLimitPolicies.

`outcallthreshold` Outcallthreshold value should be in range 0 to 10000

`outcallinterval` Outcallinterval value should be in range 1 to 300

`paramValue` List of parameter values. The following are the parameters for the ratelimit control:

- limittype - Optional value. The default value is NULL.
- devicename - Optional value. The default value is NULL.
- partition - Optional value. The default value is NULL.
- name - Optional value. The default value is NULL.
- mod - Optional value. The default value is NULL.
- layer - Optional value. The default value is NULL.
- limitrate - Limit rate.
- limitburst - limit burst
- ulograte - ULOG rate
- threshold - threshold value

`mode` Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Control Configuration File

The following are the control tracing and configuration parameters from the */nm/etc/control.config* file.

> **Important!** Editing this file is not allowed without the assistance from Alcatel-Lucent field support.

- control.logSize - Size of the log file. The default is 1,000,000.
- control.logRotate - Number of the log files. The default is 2.
- control.logFormat - Format of the log files: TEXT or XML. The default is TEXT.
- control.logLevel - The logging level. It is updated every 5 minutes. One of the following option: SEVERE, WARNING, INFO, CONFIG, ALL, OFF. The default is OFF.
- control.logConsoleLevel - The console logging level. The default is OFF.

Control Server Configuration File

The following are the udcontrol Server tracing and configuration parameters from the */nm/etc/UDCTLSRV.config* file.

> **Important!** Editing this file is not allowed without the assistance from Alcatel-Lucent field support.

- UDCTLSRV.logSize - Size of the log file. The default is 1,000,000.
- UDCTLSRV.logRotate - Number of the log files. The default is 2.
- UDCTLSRV.logFormat - Format of the log files: TEXT or XML. The default is TEXT.
- UDCTLSRV.logLevel - The logging level. It is updated every 5 minutes. One of the following option: SEVERE, WARNING, INFO, CONFIG, ALL, OFF. The default is OFF.
- UDCTLSRV.logConsoleLevel - The console logging level. The default is OFF.
- UDCTLSRV.timeOut - Timeout value in milliseconds. The default is 30000.

Error Messages

The following error:

```
control: Failed to send request to UD.
```

occurs when udaudit command could not send message to UDCTLSRV. The serve did not acknowledge the message. To solve the problem verify whetherUDCTLSRV is up and running.

The following error:

```
control: Timeout on audit response from: 'pnnxt_1'
```

occurs when udaudit did not get resposne from UDCTLSRV in a time specified in */nm/etc/UDCTLSRV.config* file. Verify whether UDCTLSRV has enough resources to execute its tasks on time.

The following error:

```
UDCTLSRV: Timeout. Did not get response from DCOL server.
```

occurs when control did not get resposne from UDNE DCOL in a time specified in */nm/etc/UDCTLSRV.config* file. Increase timeout in the configuration file or refer to UDNE DCOL documentation.

The following response:

```
DCOL: SOAP: Request does not match schema. (ratelimit)  :
   'pnnxt_3'
```

appears when error occurred in UDNE DCOL.

Examples

```
$ control type=ratelimit office=pnnxt_3 act=mod devname=MSx1
   limrate=700 part=admin limtype=ep mod=input
   name=def_ep_i_10 layer=ip
```

□

cr

Description

The `cr` (Circuit Reservation) command applies an automatic Trunk Reservation (TR) control on an outgoing or a 2-way trunk group out of a particular 1A *ESS, 5ESS, Sonus,* or 7R/E “From Office” or an automatic Circuit Reservation control on an outgoing or a 2-way trunk group out of a particular *4ESS* or *GTD-5* “From Office”.

The CR control is a Pre-Hunt 2-way trunk group control that prevents outgoing calls from using the controlled trunk group when only a specified number of trunks remain idle. Affected calls receive an announcement.

CR is a two-level control system. More severe controls are placed on outgoing traffic in the second level than in the first. Two reservation levels (thresholds), l1 and l2, are assigned to a trunk group. The reservation levels represent the number of idle trunks remaining in a trunk group. For example, if L1 = 12 and L2 = 5, the L1 controls will be activated when the number of idle trunks in the trunk group is equal to or less than 12, but greater than 5. L2 controls would be activated when the number of idle trunks in the trunk group is 5 or less.

ARA (automatic reservation adjustment) can be allowed on a CR-controlled trunk group.

- If the ARA option is not set, then the reservation levels remain fixed.
- If ARA is set, then the reservation levels are adjusted, based on the current traffic load on the trunk group.
- If there are idle trunks, ARA lowers L1 and L2 until they are zero or there are no more idle trunks.
- If there are no idle trunks, ARA raises L1 and L2, making sure they do not exceed their maximum value and that the difference between them is no greater than the difference between their maximums.

ARA eliminates the possibility that a trunk group might block “bad” traffic on the last remaining circuits in a trunk group when in fact there is no “good” traffic making bids for those circuits.

Syntax

Since the input parameters vary according to the switch and generic, the command syntax is listed for each.

Switch Type(s)	Generic(s)	Syntax

1A *ESS*	1ae8+	`cr from=`*`from`*` act=`*`action`*` tg=`*`office-suffix`*
		`[l1=`*`level-1-thresh`*`] [l2=`*`level-2-thresh`*`]`
		`[mode=`*`mode`*`]`

Switch Type(s)	Generic(s)	Syntax

4ESS	4e12+	`cr from=from act=action tg=office-suffix`
		`[ih/rs=inhibit/restore] [sk/cn=sk/cn]`
		`[cat=category] [l1=level-1-thresh]`
		`[l2=level-2-thresh] [mode=mode]`

Sonus	gsx5_1+	

4ESS	4e20 +	`cr from=from act=action tg=office-suffix`
		`[ih/rs=inhibit/restore] [sk/cn=sk/cn]`
		`[cat=category] [ara=ara] [l1=level-1-thresh]`
		`[l2=level-2-thresh] [mode=mode]`

5ESS	5e6+	
7R/E	ivtoa3, ivtoa4	

GTD-5	gtd1641+	`cr from=from act=action tg=office-suffix`
		`[sk/cn=sk/cn] [cat=category]`
		`[lvlp1=l1-percent] [lvlp2=l2-percent]`
		`[mode=mode]`

Parameters

`from` “From Office” name

`action` Required action to be taken. Valid values are:

- add = add a CR control to a trunk group in the “From Office”
- mod = modify a CR control on a trunk group in the “From Office”
- del = delete a CR control from a trunk group in the “From Office”

`office-suffix`“To Office” name plus a suffix (up to 4 characters) used to identify the trunk group the control should be placed on; separate the office name from the suffix with a hyphen (-).

`inhibit/restore` Valid values are:

- in = control inhibit
- rs = control restore/activate (default)

`sk/cn` Valid values are:

- sk = skip (default)
- cn = cancel to

`category` Response Category

Reference: Table 7

`ara` Automatic Reservation Adjustment (*4ESS*, *5ESS* and 7R/E only). Valid values are:

- Yes = Yes, ARA active
- No = No, ARA not active (default)

`level-1-thresh` Threshold Level 1

Important! For a 1A *ESS* switch, Level 1 must be > Level 2.

Reference: Table 7

`level-2-thresh` Threshold Level 2

Reference: Table 7

`l1-percent` Threshold Percentage Level 1. Valid values are: 0–100 in increments of 5

Important! For *GTD-5*, Level 1 Percent must be > Level 2 Percent.

`l2-percent` Threshold Percentage Level 2. Valid values are: 0–100 in increments of 5

`mode` Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Table

The following tables provide additional information about the parameters for this command.

Table 7 Circuit reservation control table (Sheet 1 of 2)

Variable	Generic or Switch	Values
category	*4ESS*	a, b, c, d, e, f, g, h, i, j, k, l, m, n (see Table 7 for control response category settings)
	5ESS generics 5e5 and later, 7R/E generic ivtoa3, ivtoa4, *Sonus* gsx5_1	a, b, c, d, e (see Table 9 for control response category settings)
	GTD-5	a, b, c, d, e (see Table 10 for control response category settings)
level-1-thresh	1A *ESS*	0–999, inclusive
	Sonus gsx5_1, *4ESS* and *5ESS* generics 5e14 and earlier	0–15, inclusive (see Table 8 to determine *level-1-thresh* values based on *ara* parameter values)
	5e15 and later, 7R/E generic ivtoa3, ivtoa4	0-65,534, inclusive (see Table 8 to determine *level-1-thresh* values based on *ara* parameter values)

Table 7 Circuit reservation control table (Sheet 2 of 2)

Variable	Generic or Switch	Values
`level-2-thresh`	1A *ESS*	0–999, inclusive
	Sonus gsx5_1, *4ESS* and *5ESS* generics 5e14 and earlier	0–15, inclusive (see Table 8 to determine `level-2-thresh` values based on `ara` parameter values)
	5e15 and later, 7R/E generic ivtoa3, ivtoa4	0-65,534, inclusive (see Table 8 to determine `level-2-thresh` values based on `ara` parameter values)

Table 8 Threshold levels based on ARA status

Switch Type	If ara=yes	If ara=no
4ESS	Not applicable	L1 = 0, L2 = 0 ***OR*** L2 = 0, 0 <= L1 <= 15 ***OR*** 0 < L2 < L1 <= 15
Sonus gsx5_1, *5ESS to 5ESS* generics 5e14 and earlier	1 <= L2 < L1 <= 15	0 <= L2 < L1 <= 15
5ESS generics 5e15 and later; 7R/E generic ivtoa3, ivtoa4	1<=L2<=L1<=65534	0<=L2<=L1<=65534

Table 9 5ESS and 7R/E circuit reservation control response percentages

		Response Category				
Circuit Reservation Level	**Traffic Type (See Note 1)**	**A**	**B**	**C**	**D**	**E**
L1	AR	0	0	0	100	0
	DR	0	0	0	0	0
	HA	100	100	100	100	100
	HD	0	0	75	0	75
L 2	AR	0	100	100	100	0
	DR	0	0	0	100	0
	HA	100	100	100	100	100
	HD	75	0	75	100	87.5

Table 10 GTD-5 CR control response percentages

Trunk Group Reservation Level	Traffic Type (see Note 1)	Response Category				
		A	B	C	D	E
%L1	HA	100	100	100	100	100
	HD	0	0	75	0	75
%L2	HA	100	100	100	100	100
	HD	75	0	75	100	85
	EA	0	100	100	100	0

Notes

1. Definitions for the traffic types in Table 9 and Table 10 are:
   - AR = Alternate Routed
   - DR = Direct Routed
   - EA = Easy-to-Reach, Alternate Routed
   - HA = Hard-to-Reach, Alternate Routed
   - HD = Hard-to-Reach, Direct Routed

Examples

The following is an example of a cr command for a 1A *ESS* switch:

```
$ cr from=sim1ae8 tg=sm02-sm03 act=add l1=2 l2=1
```

The following is an example of a cr command for a *4ESS* switch:

```
$ cr from=abcd tg=efgh-003 act=add ih/rs=rs sk/cn=sk cat=a
  l1=5 l2=3
```

The following is an example of a cr command for a *5ESS* switch:

```
$ cr from=d5e6 tg=sm02-003 act=add ih/rs=ih sk/cn=sk cat=d
  ara=yes l1=4 l2=2
```

The following is an example of a cr command for a *GTD-5* switch:

```
$ cr from=eriepa act=add tg=anhmca01amd-fg sk/cn=sk cat=a
  lvlp1=100 lvlp2=0
```

Related command(s)

“ctrlog” (p. 59)

□

cro

Description

The `cro` (cancel rerouted overflow) command activates or removes a CRO control on an outgoing or a 2-way trunk group out of a particular “From Office.”

A CRO control prevents overflow traffic on a via route (VB) from overflowing back to the direct route (AV). Not activating CRO can result in an external loop.

Syntax

```
cro  from=from act=action tg=office-suffix [mode=mode]
```

Switch restrictions

This command applies only to *4ESS* (4e12+), *5ESS*, and 7R/E switch types.

Parameters

`from` “From Office” name

`action` Required action to be taken. Valid values are:

- add = add a CRO control to a trunk group in the “From Office”
- del = delete a CRO control from a trunk group in the “From Office”

`office-suffix` “To Office” name plus a suffix (up to four characters) used to identify the trunk group the control is associated with; separate the office name from the suffix with a hyphen (-).

`mode` Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Examples

```
$ cro from=cleveoh4e095 act=add tg=colsoh5e003-001
```

Related command(s)

“ctrlog” (p. 59)
“total” (p. 90)

□

dptpri

Description

The dptpri (Dynamic Packet Trunks Prioritization Control) command activates, modifies, or removes a DPTPRI control on a particular “From Office”.

Syntax

```
dptpri  from=from act=action tg=office-suffix [spthr=spthr]
  [remthr=remthr] [mode=mode]
```

Feature restrictions

This command is only valid with Feature 365, “Bandwidth Directionalization & Prioritization control support in Succession Network Switch Generic SN04”.

Parameters

from — “From Office” name

action — Required action to be taken. Valid values are:

- add = add a DPTPRI control to a trunk group in the “From Office”
- mod = modify an DPTPRI control on a trunk group in the “From Office”
- del = delete a DPTPRI control from a trunk group in the “From Office”

office-suffix — “To Office” name plus a suffix (up to 4 characters) used to identify the trunk group the control is associated with; separate the office name from the suffix with a hyphen (-).

spthr — Specific TG threshold. Percentage of Terminal Identifiers (TID) that are idle. Valid values are: 1–3 digits in the range 0–100

remthr — Remaining threshold. Percentage of Terminal Identifiers (TID) that are idle for the rest of the office (Dynamic Packet Trunks (DPT) without a spthr. Valid values are: 1–3 digits in the range 0–100

mode — Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Examples

```
$ dptpri from=cleveohsn04 act=add tg=colsoh5e003-001 spthr=25
  remthr=10
```

Related command(s)

“ctrlog” (p. 59)
“total” (p. 90)

□

dptres

Description

The `dptres` (Dynamic Packet Trunks Reservation Control) command activates, modifies, or removes a DPTRES control on a particular “From Office”.

Syntax

```
dptres from=from act=action [dptthr=dptthr] [mode=mode]
```

Feature restrictions

This command is only with Feature 365, “Bandwidth Directionalization & Prioritization control support in Succession Network Switch Generic SN04”.

Parameters

`from` “From Office” name

`action` Required action to be taken. Valid values are:

- add = add a DPTRES control to a trunk group in the “From Office”
- mod = modify an DPTRES control on a trunk group in the “From Office”
- del = delete a DPTRES control from a trunk group in the “From Office”

`dptthr` DPT reservation percentage. Percentage of Terminal Identifiers (TID) that are idle for the rest of the office (OfficeThreshold). Valid values are: 1–3 digits in the range 0–100

`mode` Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Examples

```
$ dptres from=cleveohsn04 act=add dptthr=25
```

Related command(s)

“ctrlog” (p. 59)

□

dpttid

Description

The dpttid (Dynamic Packet Trunks Terminal Identifiers Control) command activates, modifies, or removes a DPTTID control on a particular “From Office”.

Syntax

```
dpttid from=from act=action [tids=tids] [mode=mode]
```

Feature restrictions

This command is only with Feature 365, “Bandwidth Directionalization & Prioritization control support in Succession Network Switch Generic SN04”.

Parameters

`from` “From Office” name

`action` Required action to be taken. Valid values are:

- add = add a DPTTID control to a trunk group in the “From Office”
- mod = modify an DPTTID control on a trunk group in the “From Office”
- del = delete a DPTTID control from a trunk group in the “From Office”

`tids` Maximum number of Terminal Identifiers (TIDs). Valid values are: 1–7 digits in the range 1–1,048,576

`mode` Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Examples

```
$ dpttid from=cleveohsn04 act=add tids=1888
```

Related command(s)

“ctrlog” (p. 59)

□

Description

The rc (routing code) command applies an access tandem routing code for a CG control.

Syntax

```
rc office=office act=action [ic=carrier ID] [rc=rc] [isrc=isrc]
   [ccd=ccd] [gap=gap] [annc=announcement] [ftype=ftype]
   [mode=mode]
```

Switch restrictions

This command is only valid for *GTD-5*.

Parameters

office — Office name

Important! You can enter multiple values separated by the plus sign (+)

action — Required action. Specify one of the following:

- add = add RC control to office
- mod = modify RC control presently on office
- del = delete RC control from office

Important! For the rc command, the following fields are optional keys. This means that if you add any of these parameters with act=add, you must also specify them for act=mod and act=del.

carrier ID — Carrier Prefix Code; enter 4 digits for the carrier prefix code

rc — Routing code. Valid values are numbers in the ranges of: 000–099 and 120–199.

isrc — ISUP Routing code. Enter 2 digits in the range 0-99.

ccd — Country code digit. 1 to 3 digits in the range 0–999

gap — Gap interval (the maximum number of calls allowed through in a 5-minute period)

Reference: Table 5, “Call gap control table” (p. 24)

announcement — Announcement type. Valid values are: nca, roa, rot, fea, sea.

ftype — b, d, or c (combined)

mode — Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Examples

```
$ rc office=gtd51711 act=add ic=0113 rc=114 gap=2.5 annc=nca
```

Related command(s)

“cg” (p. 21)
“ctrlog” (p. 59)
“total” (p. 90)

□

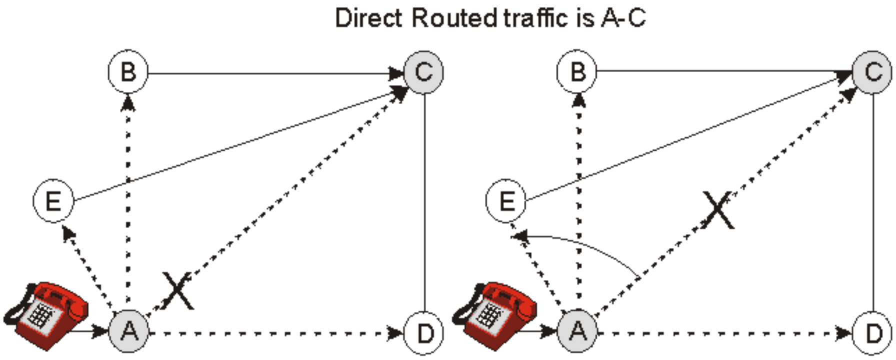
Description

The `rr` (reroute) command activates, modifies, or removes an RR control on an outgoing or a 2-way trunk group out of a particular “From Office”.

The expansive RR control modifies the normal in-chain routing by passing the call to an out-of-chain group instead of to the normal group. An in-chain trunk group is one that normally will be hunted for ordinary standard routing for a given destination. An out-of-chain trunk group is one that will normally not be hunted for routing and that has temporary idle capacity at the time of the reroute requirement.

An RR control is a Post-Hunt control that diverts affected traffic to the via routes after it overflows the controlled trunk group.

Figure 8 Graphical representation of RR control



Immediate Reroute (IRR)
IRR causes traffic to be rerouted before it is offered to a trunk group (Pre-Hunt)

- Direct A-C traffic is alternate routed to A-E
- E is an out-of-chain route

Overflow Reroute (ORR)
The ORR control diverts the affected traffic to the via route after it overflows the controlled trunk group

- Traffic A-C has no circuit available
- Reroute overflow to A-E

Syntax

Since the input parameters vary according to the switch and generic, the command syntax is listed for each.

Switch Type(s)	Generic(s)	Syntax
1A *ESS*		`rr  from=from act=action tg=office-suffix [hunt=hunt] [alt=alt] [dir=dir] [type=type] [options=options] [vias=vias] [mode=mode]`
DMS 100/200		
DMS 250		
SCSN	sn02	
LSSGR		
EWSD	ewsd13+	
Sonus	gsx5_1+	

Switch Type(s)	Generic(s)	Syntax
4ESS	4e12	rr from=*from* act=*action* tg=*office-suffix* [rrtype=*rrtype*] [type=*type*] [alt=*alt*] [dir=*dir*] [domains=*domains*] [options=*options*] [vias=*vias*] [vrto=*via turnoff override*] [code=*codes*] [rdbi=*rdbi*] [mode=*mode*]
	4e13–4e16	rr from=*from* act=*action* tg=*office-suffix* [rrtype=*rrtype*] [type=*type*] [tfc=*tfc*] [rout=*rout*] [alt=*alt*] [dir=*dir*] [domains=*domains*] [options=*options*] [vias=*vias*] [vrto=*via turnoff override*] [rate=*rate*] [code=*codes*] [rdbi=*rdbi*] [mode=*mode*]
	4e17+	rr from=*from* act=*action* tg=*office-suffix* [rrtype=*rrtype*] [type=*type*] [tfc=*tfc*] [rout=*rout*] [alt=*alt*] [dir=*dir*] [domains=*domains*] [options=*options*] [npa=*npa*] [vias=*vias*] [vrto=*via turnoff override*] [vnpa=*vnpa*] [rate=*rate*] [code=*codes*] [rdbi=*rdbi*] [mode=*mode*]
5ESS	5e4	rr from=*from* act=*action* tg=*office-suffix* [hunt=*hunt*] [alt=*alt*] [dir=*dir*] [options=*options*] [vias=*vias*] [vrto=*via turnoff override*] [mode=*mode*]
	5e5–5e14	rr from=*from* act=*action* tg=*office-suffix* [hunt=*hunt*] [alt=*alt*] [dir=*dir*] [i/x=*inc/xcl*] [domains=*domains*] [options=*options*] [vias=*vias*] [vrto=*via turnoff override*] [mode=*mode*]
	5e15	rr from=*from* act=*action* tg=*office-suffix* [hunt=*hunt*] [type=*type*] [alt=*alt*] [dir=*dir*] [i/x=*inc/xcl*] [domains=*domains*] [options=*options*] [vias=*vias*] [vrto=*via turnoff override*] [mode=*mode*]
7R/E	ivtoa3, ivtoa4	
	ivtoa4_1	rr from=*from* act=*action* tg=*office-suffix* [hunt=*hunt*] [rrtype=*rrtype*] [type=*type*] [alt=*alt*] [dir=*dir*] [i/x=*inc/xcl*] [domains=*domains*] [options=*options*] [prefix=*prefix*] [ic=*ic*] [code=*codes*] [vias=*vias*] [vrto=*via turnoff override*] [mode=*mode*]
5ESS	5e16_1	
	5e16_1h +	rr from=*from* act=*action* tg=*office-suffix* [hunt=*hunt*] [rrtype=*rrtype*] [type=*type*] [halt=*halt*] [hdir=*hdir*] [ealt=*ealt*] [edir=*edir*] [i/x=*inc/xcl*] [domains=*domains*] [options=*options*] [prefix=*prefix*] [ic=*ic*] [code=*codes*] [vias=*vias*] [vrto=*via turnoff override*] [mode=*mode*]

Switch Type(s)	Generic(s)	Syntax
SCSN	sn03+	`rr from=`*from* `act=`*action* `tg=`*office-suffix* `[hunt=`*hunt*`] [rrtype=`*rrtype*`] [halt=`*halt*`] [hdir=`*hdir*`] [ealt=`*ealt*`] [edir=`*edir*`] [type=`*type*`] [options=`*options*`] [prefix1=`*prefix*`] [ic1=`*ic*`] [digits1=`*digits*`] [prefix2=`*prefix*`] [ic2=`*ic*`] [digits2=`*digits*`] [prefix3=`*prefix*`] [ic3=`*ic*`] [digits3=`*digits*`] [prefix4=`*prefix*`] [ic4=`*ic*`] [digits4=`*digits*`] [prefix5=`*prefix*`] [ic5=`*ic*`] [digits5=`*digits*`] [prefix6=`*prefix*`] [ic6=`*ic*`] [digits6=`*digits*`] [prefix7=`*prefix*`] [ic7=`*ic*`] [digits7=`*digits*`] [prefix8=`*prefix*`] [ic8=`*ic*`] [digits8=`*digits*`] [prefix9=`*prefix*`] [ic9=`*ic*`] [digits9=`*digits*`] [prefix10=`*prefix*`] [ic10=`*ic*`] [digits10=`*digits*`] [prefix11=`*prefix*`] [ic11=`*ic*`] [digits11=`*digits*`] [prefix12=`*prefix*`] [ic12=`*ic*`] [digits12=`*digits*`] [prefix13=`*prefix*`] [ic13=`*ic*`] [digits13=`*digits*`] [prefix14=`*prefix*`] [ic14=`*ic*`] [digits14=`*digits*`] [prefix15=`*prefix*`] [ic15=`*ic*`] [digits15=`*digits*`] [prefix16=`*prefix*`] [ic16=`*ic*`] [digits16=`*digits*`] [vias=`*vias*`] [mode=`*mode*`]`
GTD-5	gtd1641	`rr from=`*from* `act=`*action* `tg=`*office-suffix* `rrtype=`*rrtype* `[halt=`*halt*`] [hdir=`*hdir*`] [ealt=`*ealt*`] [edir=`*edir*`] [cri1=`*crix*`] [nri1=`*nrix*`] [cri2=`*crix*`] [nri2=`*nrix*`] [cri3=`*crix*`] [nri3=`*nrix*`] [cri4=`*crix*`] [nri4=`*nrix*`] [cri5=`*crix*`] [nri5=`*nrix*`] [cri6=`*crix*`] [nri6=`*nrix*`] [cri7=`*crix*`] [nri7=`*nrix*`] [cri8=`*crix*`] [nri8=`*nrix*`] [mode=`*mode*`]`
	gtd1711 +	`rr from=`*from* `act=`*action* `tg=`*office-suffix* `rrtype=`*rrtype* `[rout=`*rout*`] [halt=`*halt*`] [hdir=`*hdir*`] [ealt=`*ealt*`] [edir=`*edir*`] [viatype=`*viatype*`] [vias=`*vias*`] [cri1=`*crix*`] [nri1=`*nrix*`] [cri2=`*crix*`] [nri2=`*nrix*`] [cri3=`*crix*`] [nri3=`*nrix*`] [cri4=`*crix*`] [nri4=`*nrix*`] [cri5=`*crix*`] [nri5=`*nrix*`] [cri6=`*crix*`] [nri6=`*nrix*`] [cri7=`*crix*`] [nri7=`*nrix*`] [cri8=`*crix*`] [nri8=`*nrix*`] [mode=`*mode*`]`
GSP		`rr from=`*from* `act=`*action* `tg=`*office-suffix* `[type=`*type*`] [alt=`*alt*`] [dir=`*dir*`] [options=`*options*`] [vias=`*vias*`] [mode=`*mode*`]`

Parameters

`from` “From Office” name

`action` One of the following required actions:

- add = add an RR control to a trunk group in the “From Office”
- mod = modify an RR control on a trunk group in the “From Office”
- del = delete an RR control from a trunk group in the “From Office”

`office-suffix` “To Office” name plus a suffix (up to 4 characters) used to identify the trunk group with which the control is associated; separate the office name from the suffix with a hyphen (-).

`hunt` Hunt type. Valid values are:

- For 1A *ESS*, reg, spray
- For *Sonus* gsx5_1, *5ESS* and 7R/E, reg, spray, order
- For *DMS, EWSD*, and SCSN, reg, order
  - reg = uses only one via route
  - order = causes each rerouted call to search for an idle circuit in a via trunk group in the order in which the *`vias`* are specified in the `rr` command, always beginning with the first via
  - spray = causes each rerouted call to search for an idle circuit in a via trunk group in the order in which the *`vias`* are specified in the `rr` command, beginning the search with the via following the via that carried the last rerouted call. Rerouted traffic is divided evenly among the out-of-chain trunk groups through a rotation algorithm. This has the effect of evenly distributing calls among trunk groups.

`tfc` One of the following required actions:

- alt = alternate routed traffic
- dar = direct and alternate routed traffic

Reference: Note 2

`rout` Control method for selecting controlled calls. Valid values are:

- pct = percent of calls to control (default)
- rate = maximum rate of calls to control in the 5-minute interval (not valid for *GTD-5* gtd1641 switches)

Reference: Note 2

`rate` Rate of traffic for each via entered above.

- Valid values are: all, 1, 1.5, 2, 2.5, 3, 4, 5.
- If `rout=rate`, valid values are: 1, 1.5, 2, 2.5, 3, 4, 5, 6, 8, 10, 15, 30, 45, 60

Reference: Note 1, Note 2

`alt` Percent or rate of alternate routed traffic to control.

Reference: Note 2, Note 3, Note 4

`dir` Percent or rate of direct routed traffic to control.

Reference: Note 3, Note 4

`inc/xcl` Include and exclude domains (*5ESS* only)

- inc = Include (default)
- xcl = Exclude

`rrtype` Type of reroute applied. Valid values:

- *4ESS*: reg, code, rdb
- *GTD-5*: orr, irr
- *DMS,* SCSN, and 5e16_1 and later: reg, code

`codes` NPA-NXX codes if `rrtype=code`. Valid values are:

- *4ESS*: up to 16 digits
- *5ESS* and 7R/E: up to 10 digits

Reference: Note 1

`rdbi` Up to 16 RDBIs in the range 1–8191, if `rrtype=rdbi` (*4ESS* only)

Reference: Note 1

`type` Application of control relative to trunk hunt. Valid values are:

- orr = trunk group post-hunt overflow reroute. This takes the overflow traffic on a trunk group and reroutes it to a trunk group with idle capacity.
- irr = trunk group pre-hunt immediate reroute (not valid for 5e14 and earlier). This causes a percentage of traffic to be rerouted before it is offered to the regular in-chain trunk group.

`domains` List of up to 5 domains or all (default).

Important! For the *5ESS* switch, the all value is applicable only when `i/x=inc`.

Reference: These values are defined in the Record Base. See "TYPXREF File" (p. 105) and "Trunk Group Threshold File" (p. 98) in the *Record Base Administration Guide*.

`options` Reroute options.

If a *DMS* or an LSSGR switch office is specified in the from office field, the *`options`* field ***must*** include either "ea" or "nea". You may also include any other options value in addition to this.

Example: "ea", "nea", "ea+nea", "ea+nea+cicr"

Nonexample: "cicr+htr"

Reference: Note 1, Table 11

`prefix` Valid values are:

- nanp = national code

- itu = international code

  Reference: Note 5

`ic` IC prefix in the range 0001 - 9999. If this value is:

- Defaulted, a corresponding *`digits`* value must be entered.
- Entered, the *`digits`* value may or may not be entered (it can be defaulted).

  Reference: Note 5

`digits` Valid values are based on the value of *`prefix`*, as follows:

- For SCSN switch types, valid values are:
  - nanp = 0–10 digits
  - itu = 0–15 digits
- For 5e16_ 1 and later switch types, valid values are:
  - nanp = 0–7 digits
  - itu = 0–10 digits

  Reference: Note 5

`vias` Via office name plus a suffix (up to 4 characters) used to identify the via trunk group for the control; separate the office name from the suffix with a hyphen (-). Up to 7 via offices can be specified, separated by the plus sign (+).

Reference: Note 6, Note 7

`crix` Current route index number *`x`*

You can enter multiple values separated by the plus sign (+).

Reference: Note 6

`nrix` New route index number *`x`*

You can enter multiple values separated by the plus sign (+).

Reference: Note 6

`via turnoff override` Via route turnoff override. Valid values are:

- no = via route turnoff to function as normal (default)
- yes = via route turnoff to be temporarily turned off

  Reference: Note 1

`viatype` Type of vias listed (*GTD-5* only)

- index = vias are referenced by index (##.###)
- tg = vias are referenced by trunk group (*`vias`*) (not valid for *GTD-5* - 1641)

`halt` Percent or rate of hard-to-reach alternate routed traffic to control.

Reference: Note 8, Table 11

`hdir` Percent or rate of hard-to-reach direct routed traffic to control.

Reference: Note 8, Table 11

`ealt` Percent or rate of easy-to-reach alternate routed traffic to control.

Reference: Note 8, Table 11

`edir` Percent or rate of easy-to-reach direct routed traffic to control.

Reference: Note 8, Table 11

`npa` Numbering Plan Area code

`vnpa` NPA prefix indicator for the via route. The *`npa`* parameter must be specified for the *`vnpa`* parameter to be specified.

Reference: Note 1

`mode` Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Notes

1. You can enter multiple values separated by the plus sign (+).
2. When:
   - `rout=rate` — the *`rout`* and *`rate`* parameters are linked, and both must be specified.
   - `rout=pct` — the *`tfc`* and *`alt`* parameters are linked, and both must be specified.
3. The *`alt`* and *`dir`* parameters are linked keywords.
   - For all switch types except *4ESS* and *GTD-5* switches, you must specify *`alt`*, *`dir`*, or both, for the control to function. The values entered can be different.
   - For *4ESS* switches prior to 4e17, the *`dir`* parameter cannot be specified alone. When you desire to control all traffic, enter the same value for both the *`alt`* and *`dir`* parameters. If a different value is entered, the value in the *`alt`* parameter will be used as the value to control both alternate and direct routed traffic.
   - In 4e13 and later switches, the *`alt`* and *`dir`* parameters are linked to the *`tfc`* parameter.
   - For 4e17 and later switches, specifying `tfc=dir` will allow the *`dir`* parameter to be specified alone. If `tfc=dar` is selected, you must enter a value in both the *`alt`* and *`dir`* parameters, but, as in previous generics, the *`alt`* value will be used for the control and the *`dir`* value will be ignored.
4. Valid values for *`alt`* and *`dir`* are:

- For generics other than 5e15 and later, and ivtoa3 and later: 0, 12, 25, 37, 50, 62, 75, 87, 100
- For 5e15 and later, and ivtoa3 and later: 0–100, inclusive

Important! *4ESS* switches do not allow a value for alt or dir of 0.

5. The `prefix`, `ic`, and `digits` parameters are linked keywords.
   - Up to 16 combinations may be entered.
   - At least one must be used if `rrtype=code`.
6. The `vias`, `crix`, and `nrix` parameters are linked keywords. You must specify either a `vias` parameter or paired `crix` and `nrix` parameters (i.e. if a `crix` parameter is specified, an `nrix` with at least one value must be specified for the control to function).
7. The GTD/5 switch generics 1711 and later support the `viatype=tg` and `vias` parameters. In order to use this control from NTM, it is necessary to set up a routing table for the trunk group in the GTD/5 switch. The commands on the GTD/5 switch are:
   - `ADD VRTB` - this command adds a via route table
   - `SET NSM NTGR` - this command assigns route indexes to the reroute trunk and via trunk

For more information on these commands, see the GTD/5 documentation.

8. At least one of the `halt`, `hdir`, `ealt` and `edir` parameters must have a non-zero value for the control to function.

Table

The following table provides additional information about the parameters for this command.

Table 11 Reroute control table

Variable	Generic or Switch	Values
`options`	1A *ESS*	cicr
	4ESS	htr, nhr, apr, cicr, piit
	5ESS, 7R/E	apr, cicr, piit
	DMS, SCSN, *Sonus* gsx5_1, GSP gsp07+	cicr, nea, ea, htr
	EWSD	cicr, nea, ea

`halt`
`hdir`
`ealt`
`edir`
	GTD-5	0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100

	ivtoa 4_1 +, 5e16_1 h+	0–100 in increments of 1

Table 11 Reroute control table (continued)

Variable	Generic or Switch	Values
(if `rout=rate`)	*GTD-5* - 1711, 1721, 1722	0–60 in increments of 1, 120, 300, 600, p5, inf

Value acronyms mean:

- **RETURN** = no options specified
- apr = allow previously rerouted
- cicr = cancel in-chain return
- ea = equal access
- htr = hard-to-reach
- nea = Non-equal access
- nhr = non hard-to-reach
- piit = prohibit international inbound traffic

Examples

The following is an `rr` command line to activate a reroute in a *DMS*, LSSGR, or 1A *ESS* switch:

```
$ rr from=dms24 tg=sm02-sm03 act=add hunt=reg alt=25 dir=37
   type=orr options=ea+cicr vias=abc-005
```

The following is an `rr` command line for a rate-based code reroute for a *4ESS* switch, generic 4e13:

```
$ rr from=abc tg=abc-004 act=add rrtype=code tfc=alt rout=rate
   type=irr options=htr+cicr vias=abc-005+abc-004 vrto=yes+no
   rate=3+10 domains=pots codes=459623+614860
```

This is an `rr` command line for a reroute for a *5ESS* switch, generic 5e5:

```
$ rr from=d5e5 tg=sm02-003 act=add hunt=reg alt=25 dir=12
   options=cicr vias=barcln5x-003 vrto=yes i/x=inc dom=all
```

The following is an `rr` command line for a percent-based route index reroute for a *GTD-5* switch, generic gtd1641:

```
$ rr from=eriepa act=add tg=anhmca01amd-fg rrtype=orr halt=50
   hdir=50 ealt=50 edir=50 crix1=0.99 nrix1=10.200 crix1=0.99
   nrix1=10.200 crix2=0.99 nrix1=10.200 crix3=0.99
   nrix1=10.200 crix4=0.99 nrix1=10.200 crix5=0.99
   nrix1=10.200 crix6=0.99 nrix1=10.200 crix7=0.99
   nrix1=10.200 crix8=0.99
```

The following is an `rr` command line for a percent-based route index reroute for a *GTD-5* switch, generics gtd1711, gtd1721, and later:

```
$ rr from=anhmcapx40t act=add tg=snmncaxp53t-c7f rrtype=orr
   rout=pct halt=100 ealt=100 hdir=100 edir=100 viatype=index
   cri1=0.035 nri1=0.085
```

Related command(s)

“ctrlog” (p. 59)
“total” (p. 90)

□

silc

Description

The silc (selective incoming overload control) command activates or removes a SILC control on an incoming or a 2-way trunk group in a particular “From Office”.

SILC is an automatic incoming trunk group control that occurs at the switch and is administered by both switch and NTM personnel. SILC restricts incoming calls to a switch. Depending on the switch, a percentage or a measured rate of calls is affected and receives the Overflow Signal.

SILC blocks set percentages of incoming traffic on the trunk groups to which it is assigned when the switch processor enters either MC1 (machine congestion level 1) or MC2 (machine congestion level 2) level of processor overload. When the overload is reduced below MC1 level, SILC blocking action on incoming traffic automatically stops. The two percentages of calls blocked by SILC at MC1 and MC2 are specified by the specific switch type and applied to all SILC-assigned trunk groups in the office. The percentages are not controllable from NTM.

SILC can be enabled/disabled (allowed/inhibited) either on-site by switch personnel or from NTM. Assignment of trunks to the SILC control list or deletion from the control list is done only at the switch.

Syntax

```
silc  from=from act=action tg=office-suffix [mode=mode]
```

Switch restrictions

This command only applies to 1A *ESS, 5ESS*, 7R/E, and *GTD-5* switch types.

Parameters

from “From Office” name

action Required action to be taken. Valid values are:

- enable = enable SILC on a trunk group in the “From Office”
- disable = disable SILC on a trunk group in the “From Office”

office-suffix “To Office” name plus a suffix (up to 4 characters) used to identify the trunk group the control is associated with; separate the office name from the suffix with a hyphen (-).

mode Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Examples

```
$ silc from=dllstxg0524 act=enable tg=ftwttxg0132-001
```

Related command(s)

"ofcovrd" (p. 68)

□

Control-related commands

Overview

Purpose

Control-related commands display or modify information about the control commands or provide additional data for support in applying controls.

Table

Table 12 lists each NTM control-related command and its function.

Table 12 Control-related commands and functions

Command	Function	Office / Feature Restrictions
`ctrlactlist`	Retrieves all active TG controls and Call Gap controls	Feature 272, "NTM Report Writer"
`ctrlog`	Lists active or matched controls for all control types	
`dhtr`	Applies or removes domestic HTR (Hard-To-Reach) codes	*4ESS, 5ESS,* 7R/E, *GTD-5, SCSN*
`ihtr`	Applies or removes international HTR codes	*4ESS, 5ESS* (5e16_1 and later), 7R/E, SCSN
`ofcovrd`	Removes a particular type of control from a specified office	*5ESS*
`pp`	Applies or removes a preprogrammed trunk group control	1A *ESS*
`pplist`	Lists all preprogram controls that can be activated or deactivated	1A *ESS*
`preplan`	Edits or implements a control preplan	
`purglog`	Removes matched control entries from the database	
`total`	Removes all instances of a class of controls from an office	

□

ctrlactlist

Description

The `ctrlactlist` command retrieves all active TG controls and Call Gap controls in case the Reporting Host loses all control data. It also forwards these controls to the `ctrldump` to form SQL script files for updating the Reporting Host.

Syntax

```
ctrlactlist
```

Important! No other input parameters are required.

Feature restrictions

This command is only valid if you have purchased Feature 272, “NTM Report Writer”.

System responses

Figure 9 shows an example of the type of message displayed when the total command is successfully executed.

Figure 9 ctrlactlist command output

```
**** ACTIVE CONTROL(S) __ OUTPUT SORTED ****

ON: 98/11/19 14:42:30 BY: AUDIT OFF:
cg office=elcnca01ds0 act=add code=976555       gap=1200
dom=all annc=ea1

TOTAL NUMBER ACTIVE CONTROL(S) is 1
```

□

ctrlog

Description

The ctrlog (control log) command provides a list of active or matched controls for all control types. (Matched controls are controls that have been added and then deleted.)

The control log is purged automatically once a week to remove all matched controls that are more than a month old. Before the automatic purge is performed, all matched controls in the control log are saved to the “/musr/log/ctrlog.out” file.

The recommended capacity limit for the control log is dependent on the NTM configuration purchased. For limitations, see Chapter 10, “NTM Engineering Guidelines” in the *System Overview* guide.

Syntax

```
ctrlog office=office type=type status=status [trap=trap]
   [by=login] [start=start] [stop=stop]
```

Parameters

office	Office name or all
type	Control type:

- acc = automatic congestion control
- canf = cancel from
- cant = cancel to
- cg = call gap
- cr = circuit reservation
- cro = cancel reroute overflow
- dptpri = dynamic packet trunks prioritization
- htr = hard-to-reach (Automatic htr controls are not logged)
- rr = regular reroute
- skip = skip trunk group
- all = all control types

Notes:

1. Not all listed controls are valid for all switch types.
2. You can enter multiple values separated by the plus sign (+).

status	Status of control log entries:

- all = active and matched

- active = currently active
- matched = no longer active
- summary = reports total number of controls in the database

Hint: If you set the type to be cant, canf, skip, or rr, and you set status to be summary, you will see a summation of all of the CANT, CANF, SKIP, or RR controls.

`trap` Call security trap

- off
- on
- all

`login` Login of person who applied controls or “all”

`start` Start time for an interval of controls

`stop` Stop time for an interval of controls

Using ctrlog

Enter the start time and stop time in the format shown below. The minimum you must enter is the minute. The remaining parameters (current hour/day/month/year) are defaulted.

To prevent defaulting to the current day, month, or year, you must specify each of the above parameters. For example, if you enter `95/07/22-10:00` for the start time, you will receive a listing of controls starting on July 22, 1995, at 10:00 a.m.

If you enter `21-14:30` for the start time, you will receive a listing of controls starting on July 21, 1995, at 2:30 p.m. (assuming that the current month is July 1995).

> **Important!** Start and stop time define a window of time. Any control that is activated during any portion of time within the window will be listed, assuming all other parameters match.
>
> If a purglog has removed all matched controls for the specified start/stop time then you will not see those controls.

Figure 10 shows 3 active controls, all activated in different time windows.

Figure 10 ctrlog command output — different time windows

```
$ ctrlog testgtd5 all act
  IP

     **** ACTIVE CONTROL(S) -- OUTPUT SORTED ****


ON: 95/01/31 09:26:25  BY: jld     OFF:
```

```
rr    from=testgtd5 tg=anhmca0177k-ph act=add
        rrtype=orr rout=pct halt=50 hdir=40 ealt=30
        edir=20
        crix1=12.240 nrix1=34.104


ON: 95/01/05 13:27:49  BY: AUDIT     OFF:
cg    office=testgtd5 act=add ic=4321 code=1234614555
        ccd=201 pccd=2   gap=2.5 pfxt=any ftype=d


ON: 95/03/02 10:14:15  BY: nlm     OFF:
cg    office=testgtd5 act=add ic=5555 code=614871
        gap=60 pfxt=any ftype=b annc=nca

     TOTAL NUMBER ACTIVE CONTROL(S) is  3
```

To see all controls activated between Jan. 5 at 12:00 and Jan. 6 at 12:00 for the office shown in Figure 10 (testgtd5), use the command line shown in Figure 11.

Figure 11 ctrlog command output — Example 2

```
$  ctrlog office=testgtd5 type=all status=act start=95/01/05-12:00
            stop=95/01/06-12:00
  IP

     **** ACTIVE CONTROL(S) -- OUTPUT SORTED ****


ON: 95/01/05 13:27:49  BY: AUDIT     OFF:
cg    office=testgtd5 act=add ic=4321 code=1234614555
        ccd=201 pccd=2   gap=2.5 pfxt=any ftype=d


     TOTAL NUMBER ACTIVE CONTROL(S) is  1
```

System responses

The following command line can be used to generate the output shown in Figure 12.

```
$ ctrlog office=sim4e14ent03 type=all status=summary
```

Figure 12 ctrlog command output

```
IP

****    8 Active  CG  control(s)    ****
****    7 Matched CG  control(s)    ****
****   15 Total   CG  control(s)    ****
****    5 % CG  CAPACITY USED       ****
```

```
****    0 Active  ATG control(s)    ****
****    0 Matched ATG control(s)    ****
****    0 Total   ATG control(s)    ****
****    0 % ATG CAPACITY USED       ****

****   99 Active  TG  control(s)    ****
****    0 Matched TG  control(s)    ****
****   99 Total   TG  control(s)    ****
****    1 % TG  CAPACITY USED       ****

****    9 Active  HTR control(s)    ****
****    0 Matched HTR control(s)    ****
****    9 Total   HTR control(s)    ****
****    0 % HTR CAPACITY USED       ****
```

The summary option is useful to determine if a purglog is needed to create more room in the database. NTM control performance is greatly affected by the number of controls in the database.

The summary option is also useful for identifying discrepancies between the number of total controls in the database and the number of active controls. If such a discrepancy exists, please contact your system administrator or customer support representative.

Examples

This command line can be used to generate the output shown in Figure 13.

```
$ ctrlog office=abc type=cg status=active
```

Figure 13 Output from ctrlog entry

```
ON: 06-30-94 17:36:25 BY: xyz OFF:
cg office=abc code=459 act=add gap=100
annc=ea1 type=pots tg=abc-004 trap=off
```

When to use the control log summary

The control log summary is useful when you want to determine when the `purglog` command should be executed. System performance is reduced with an increase in the number of controls in the database, regardless of whether they are active or matched.

The control log summary provides a quick and accurate account of the controls in the database. If any of the four areas of the control database exceeds 75% capacity (as determined by the number of controls in the database), the summary recommends that you execute the `purglog` command.

Control log capacities

For performance considerations, follow the recommended control log limits based on the NTM configuration purchased. For limitations, see Chapter 10, “NTM Engineering Guidelines” in the *System Overview* guide.

Using ctrlog command to create a basic preplan

When you enter the `ctrlog` command, you can redirect the output into a file. For example:

```
ctrlog abc cant active > abc.cant
```

Where *abc* = office name

This provides you with a list of all active CANT controls on office *abc*. The new file, *abc*.cant, can be used to create a preplan that removes all CANT controls on office *abc*.

Viewing multiple screen output

The *Linux* system `pg` command allows you to view the output of a command one screen at a time. In the case of the `ctrlog` command, you can view all active controls a page at a time by piping the output as shown in the following:

```
ctrlog abc all active | pg
```

Where *abc* = office name

The `pg` option prevents the output from scrolling off the screen. To go on to the next page of output, press **RETURN**.

Related command(s)

“purglog” (p. 89)

□

dhtr

Description

The `dhtr` (domestic hard-to-reach) command allows you to apply a domestic HTR code to or remove it from the control or source list for one or more offices.

Syntax

Since the input parameters vary according to the switch and generic you are using, the command syntax is listed for each.

Switch Type(s)	Generic(s)	Syntax

4ESS	4e12+	`dhtr office=`*`office`*` act=`*`action`*` npa=`*`npa`*` [nxx=`*`nxx`*`]`
		`list=`*`list`*` [mode=`*`mode`*`]`

GTD-5	gtd1641+	`dhtr office=`*`office`*` act=`*`action`*` [ic=`*`ic`*`] [npa=`*`npa`*`]`
		`[nxx=`*`nxx`*`] ftype=`*`ftype`*` [mode=`*`mode`*`]`

SCSNSN	sn03+	`dhtr office=`*`office`*` act=`*`action`*` [ic=`*`ic`*`] [code=`*`code`*`]`
		`[mode=`*`mode`*`]`

5ESS	5e16_1h+	

Parameters

`office` Office name

You can enter multiple values separated by the plus sign (+).

`action` Required action. Valid values are:

- add = apply code to control or source list
- del = remove code from control or source list

`ic` Interexchange Carrier — must be a 4-digit number

`code` 0–10 digits

`npa` Numbering Plan Area (3 digits)

`nxx` Office (3 digits) or `none`.

`list` List type. (*4ESS* only) Valid values are:

- clm = HTR manual control list
- cli = HTR inhibit control list
- slm = HTR manual source list
- sli = HTR inhibit source list

`ftype` Feature group (b, d, or c [combined]) `mode` Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Examples

```
$ dhtr office=clevoh4e003 act=del npa=216 nxx=494 list=slm
```

Related command(s)

“ctrlog” (p. 59)
“ihtr” (p. 66)

□

ihtr

Description

The ihtr (international hard-to-reach) command allows you to apply an international HTR code to or remove it from a control list for a particular office(s).

Syntax

Since the input parameters vary according to the switch and generic you are using, the command syntax is listed for each.

Switch Type(s)	Generic(s)	Syntax
4ESS	4e12 +	ihtr office=*office* act=*action* cc=*cc* [nn=*nn*] list=*list* [mode=*mode*]
SCSNSN	sn03 and later	ihtr office=*office* act=*action* [ic=*ic*] [code=*code*] [mode=*mode*]
5ESS	5e16_1h and later	

Parameters

office Office name

Important! You can enter multiple values separated by the plus sign (+).

action Required action to be taken. Valid values are:

- add = add HTR code to control list
- del = delete HTR code from control list

cc Country code (1 to 3 digits)

nn National number (1 to 3 digits) or none.

ic Interexchange Carrier — must be a 4-digit number

code 0–15 digits

list List type:

- clm = HTR manual control list
- cli = HTR inhibit control list

mode Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Examples

```
$ ihtr office=clevoh4e003 act=del cc=24 nn=14 list=cli
```

Related command(s)

“ctrlog” (p. 59)
“dhtr” (p. 64)

□

ofcovrd

Description

The ofcovrd (office override) command inhibits a control of the specified *type* from a specified office.

Syntax

```
ofcovrd  office=office type=type [act=action] [mode=mode]
```

Switch restrictions

This command is valid for *5ESS* office(s) only.

Parameters

office Office name

type Control type. Valid values are:

- For 5e4 and 5e5, silcovrd
- For 5e6 and 5e9 or later, silcovrd or str_inh

action on/off

mode Execution Mode:

- exec = execute the control; this is the default
- validate = validate the control

System responses

There is no system response for this command.

Examples

```
$ ofcovrd office=colsoh5e045 type=str_inh act=on
```

Related command(s)

“ctrlog” (p. 59)

□

pp

Description

The `pp` (preprogram) command applies a predefined pp trunk group control on a particular 1A *ESS* "From Office." A preprogram control can be one of the following trunk group controls: CANF, CANT, SKIP, RR, or IRR. The specific control to be applied is determined by the preprogram number, which identifies a particular trunk group control (for example, preprogram control #1 might be a CANF control, which would be applied to the "From Office" where the `pp` command is executed, where pp=1).

Syntax

```
pp from=from act=action pp=pp [mode=mode]
```

Switch restrictions

This command is only valid for 1A *ESS* offices.

Parameters

`from` "From office" name (must be a 1A *ESS* office)

`action` Action to be taken:

- add = add the preprogram control to an office
- del = delete the preprogram control from an office

Notes:

1. Delete will also remove the inhibit option if enabled.
2. Preprogram controls applied automatically by the switch cannot be deleted.

`pp` A "Preprogram Number" that identifies a specific trunk group control previously defined within the "From office" and stored in the NTM database.

`mode` Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Examples

```
$ pp from=sim1ae8 pp=5 act=add
```

Related command(s)

"pplist" (p. 71)

□

pplist

Description

The `pplist` (preprogram control list) command lists all the preprogram controls that can be activated or deactivated for one or more 1A *ESS* offices. Preprogram controls are switch-defined trunk group controls (CANF, CANT, SKIP, RR, or IRR), records of which are stored in the NTM database. Preprogram control information displayed by the `pplist` command is retrieved from the NTM database.

Syntax

```
pplist office=office
```

Parameters

`office` Office name (must be a 1A *ESS* office) or `all`

Important! You can enter multiple values separated by the plus sign (+).

System responses

Figure 14 shows the responses for the `pplist` command.

Figure 14 pplist system responses

IF the command is ...	THEN the system responds with the ...
executed successfully	IP followed by list of available PP controls
not executed successfully	appropriate error message

Examples

The following is an example of a command to list the preprogram controls for a particular 1A *ESS* switch:

```
$ pplist office=sim1ae8
```

The following is an example of a command to list the preprogram controls for all 1A *ESS* switches:

```
$ pplist office=all
```

Related command(s)

"pp" (p. 69)

□

preplan

Description

The preplan command is used with preplan files. A preplan is a file that consists of a collection of manual network management controls.

Syntax

```
preplan name=name act=action
```

Parameters

name	Preplan name
action	Required action to be taken. Valid values are:

- add = add controls of preplan
- del = delete controls of preplan
- edit = edit the preplan
- vld = validate the preplan
- destroy = destroy entire preplan — preplan will no longer exist
- quit = exit preplan command

Detailed description

You can build preplan files using a standard *Linux* system editor and you can execute them directly from the shell prompt, but this is not recommended because it bypasses the built-in protections provided by the preplan command.

These protections include validation of a preplan before execution and proper execution of the preplan. If a validated version of the preplan file does not exist, the system prints an error message stating that the preplan has not been validated, and the execution process terminates.

The add and del preplan actions affect controls on the network. The add action results in executing the controls in the designated preplan file on the network. The del action results in removing the controls in the file from the network.

The other preplan actions are concerned with creation and maintenance of the preplan file itself.

- The edit action brings up an editor so that you can create a new preplan file or modify an existing one in the "/musr/preplans" directory. You can copy control information from the ctrlog display into your preplans file.

- The `vld` action checks all commands in the preplan file to make sure they have valid control command and shell programming syntax. Preplans that have passed the validation test are marked by the ".vld" extension.
- The `destroy` action removes the preplan file from the "/musr/preplans" directory.
- The `quit` action exits the preplan command and returns the shell prompt to the screen.

System responses

Figure 15 shows the responses for the `preplan` command.

Figure 15 preplan system responses

IF the command is ...	THEN the system responds with ...
executed successfully	IP
not executed successfully	the appropriate error message
is executed, but the preplan has not been validated	?E INVALID PREPLAN NAME: *name* is invalid; it has not been validated.
is executed, but the preplan fails	NG PREPLAN *name* WAS UNSUCCESSFUL
is executed with the `action=edit` option, and you make changes to the preplan	OK - *name*: HAS BEEN CHANGED
is executed with the `action=edit` option, and you do not make changes to the preplan	OK - *name*: HAS NOT BEEN CHANGED
is executed with the `action=vld` option	*name* VALIDATING CONTROL COMMAND SYNTAX (Using control command parameter validation) OK VALIDATING Linux SHELL PROGRAMMING SYNTAX (using /bin/ksh -nue /musr/preplans/*name* add) OK (using /bin/ksh -nue /musr/preplans/*name* del) OK

Figure 15 preplan system responses

IF the command is ...	THEN the system responds with ...

is executed with the `action=destroy` option	*name*?
	You are prompted for a confirmation before the preplan is removed.
	• If you respond **y** (yes), the system responds with:
	`Removing plan` *name*
	`OK -` *name*`: DESTROYED`
	• If you do not respond **y**, the system responds with:
	`OK -` *name*`: NOT DESTROYED`

Examples

The following example executes a preplan named preplan1 with add as the action being taken.

```
$ preplan name=preplan1 act=add
```

Copying information from ctrlog

Information in the control log can also be used to create a preplan. This allows the user to take a short cut to build a preplan using existing information. The output from control log entries is stored in the "/musr/log/ctrlog.out" file. Copy the output from a Control Log entry into a new file under the preplan directory ("/musr/preplans"). You can then make minor changes to this file to create a preplan.

Figure 16 shows an existing Control Log entry that has been copied to a new file. Figure 17 shows the resulting Preplan after this file has been edited.

Figure 16 File containing control log output

```
ON: 06-30-93 17:36:25 BY: xyz OFF:
cg office=abc code=459 act=add gap=100
annc=ea1 type=pots tg=abc-004 trap=off
```

To create a preplan from this file, make the following changes:

1. Delete the first line: `ON: 06-30-93 17:36:25 BY: xyz OFF:`.
2. Change `act=add` to `act=${1}`.
3. If the control command takes more than one line, end intermediate lines with a space and backslash (\). This allows the command to be treated as one continuous line.

   Important! There must not be a space between the backslash and the return at the end of the line.
4. Allow only one space between fields.

Figure 17 shows the new preplan created from the control log entry.

Figure 17 Preplan created from control log entry

```
cg office=abc code=459 act=${1} gap=100 \
annc=ea1 type=pots tg=abc-004 trap=off
```

Basic preplans

A ***basic preplan*** is a file that:

- Consists of a collection of network management commands (usually control commands)
- Executes these commands for your particular network situations, such as switch failure and peak holiday traffic scenarios

You can create an unlimited number of preplans.

> **Important!** The actual preplan files are stored in the preplans directory "/musr/preplans." Do not use this directory to store anything except the preplan files.

Creating basic preplans

Use the `preplan` command to create all basic preplans from the *Linux* system shell prompt. You can also use the `preplan` command to create and change preplan scripts (*Linux* system shell programs) containing NTM control commands.

The `preplan` command uses the editor defined in your *Linux* system shell environmental variable "EDITOR." You can set "EDITOR" to the "vi" (or "ed") text editor by entering:

```
EDITOR=vi; export EDITOR
```

from the *Linux* system shell prompt. The "/bin/ed" line editor becomes the default when an editor is not defined.

Preplan file names

The name of the files containing the preplan must:

- Adhere to all standard *Linux* system file-naming conventions
- Be a minimum of 2 and a maximum of 10 characters
- ***Not*** contain the plus sign (+)

Preplan actions

Figure 18 describes the actions you can perform on a preplan.

Figure 18 Preplan actions

IF you want to ...	THEN use the ... option
create a preplan or change an existing one	edit

Figure 18 Preplan actions

IF you want to ...	THEN use the ... option
validate that the information in a preplan has valid syntax	vld (validate)
implement the preplan to add controls to the network	add
implement the preplan to remove controls from the network	del (delete)
remove the preplan file from the "/musr/preplan" directory	destroy
exit from the `preplan` command	quit

The `add` and `del` preplan actions affect controls on the network. The `add` action results in executing the controls in the designated preplan file on the network. The `del` action results in removing the controls in the file from the network.

The other preplan actions are concerned with creation and maintenance of the preplan file itself.

- The `edit` action brings up an editor so that you can create a new preplan file or modify an existing one in the "/musr/preplans" directory. You can copy control information from the `ctrlog` display into your preplans file.
- The `vld` action checks all commands in the preplan file to make sure they have valid control command and shell programming syntax. Preplans that have passed the validation test are marked by the ".vld" extension.
- The `destroy` action removes the preplan file from the "/musr/preplans" directory.
- The `quit` action exits the `preplan` command and returns the shell prompt to the screen.

The following sections provide more detail about these actions.

Editing the preplan

Use the edit option with the `preplan` command to create a new preplan or change an existing one.

Using control commands within preplans

Figure 19 shows the general command syntax for NTM control commands within a preplan.

Figure 19 General control command syntax in a preplan

```
command name keyword=value(s) . . . keyword=value(s)
```

In Figure 19:

- *command name* is the name of the NTM control command (for example, cg, rr, or total)
- *keyword* is the descriptive name of the keyword value(s) to follow (for example, office name, type, action, etc.)
- *value(s)* is the numeric or string value appropriate for that keyword

Notes:

1. Preplans accept only name-defined parameters (as shown in Figure 19). Position-defined parameters are not accepted for preplans (for example: cg snjn,add,6142318700,5,nca,all).
2. Required keywords and values may vary, depending on the switch type and/or generic. If you are uncertain of the proper syntax for a control, enter the control command followed by a space and a question mark (?) to request the valid syntax. For example, enter `cg ?` for the valid syntax of the `cg` command.

Guidelines for creating preplans

When you create a preplan, you must:

- End each control command with a newline character.

  Important! In cases where, for readability, control commands require more than 1 line, end intermediate lines with a space and a backslash (\). The control command is then treated as one continuous line during execution. Precede each backslash with a space.

- Begin each control command name on a new line within the preplan.
- Use the *Linux* system shell programming parameter substitution for the `act` keyword of each control command.

  The run-time variable `${1}` is a shell variable that allows users to execute commands in a preplan file with either the `add` or the `del` action. Users enter the `add` or `del` action at the `action=` prompt when executing the preplan file in the prompted mode, or they type it on the command line as an argument to the `preplan` command, for example:

  ```
  preplan plan1 add
  preplan plan1 del
  ```

  Once you have finished editing, you can exit the editor by using normal procedures for your editor. The preplan prompt new `action=` comes up. At this prompt you can enter another action parameter value, such as `vld` or `quit`.

- The `preplan` command action values of add and del will be passed as ${1}. No other preplan parameter substitutions are allowed in a preplan file.

Example: Basic Preplan

Figure 20 shows a basic preplan for a *4ESS* switch office.

Figure 20 Sample basic preplan — 4ESS office

```
rr from=ofc_A tg=ofc_B-004 act=${1} rrtype=code tfc=alt rout=rate \
options=htr+cicr vias=ofc_C-006+ofc_D-003 vrto=yes+no \
rate=3+10 domains=pots+orig codes=459+614

cg office=ofc_B act=${1} code=7220735 gap=20 \
annc=ea2 type=pots tg=none trap=off
```

Creating preplans by copying information from the control log

Information in the control log can also be used to simplify the task of creating a preplan file. This allows the user to take a shortcut to build a preplan using existing information. The output from the `ctrlog` command is stored in the "/musr/log/ctrlog.out" file. Copy the output from a control log entry into a new file under the preplan directory ("/musr/preplans"). You can then make minor changes to this file to create a preplan.

Figure 21 shows an existing control log entry that has been copied to a new file. Figure 22 shows the resulting preplan after this file has been edited.

Figure 21 Partial ctrlog output

```
ON: 08-01-94 17:36:25 BY: xyz OFF:
cg office=abc code=459 act=add gap=100
annc=eal type=dom tg=abc-004 trap=off
```

To create a preplan from this file, make the following changes:

1. Delete the first line:

```
ON: 08-01-94 17:36:25 BY: xyz OFF:
```

2. Change `act=add` to `act=${1}`.
3. If the control command takes more than 1 line, end intermediate lines with a space and a backslash (\). This allows the command to be treated as 1 continuous line.

   Important! There must be no space between the backslash and the return at the end of the line.

4. Allow only 1 space between fields. Remove any extra spaces.

Figure 22 shows the new preplan created from the control log entry.

Figure 22 Creating a preplan from a control log entry

```
cg office=abc code=459 act=${1} gap=100 \
annc=eal type=pots tg=abc-004 trap=off
```

Reference: "ctrlog" (p. 59)

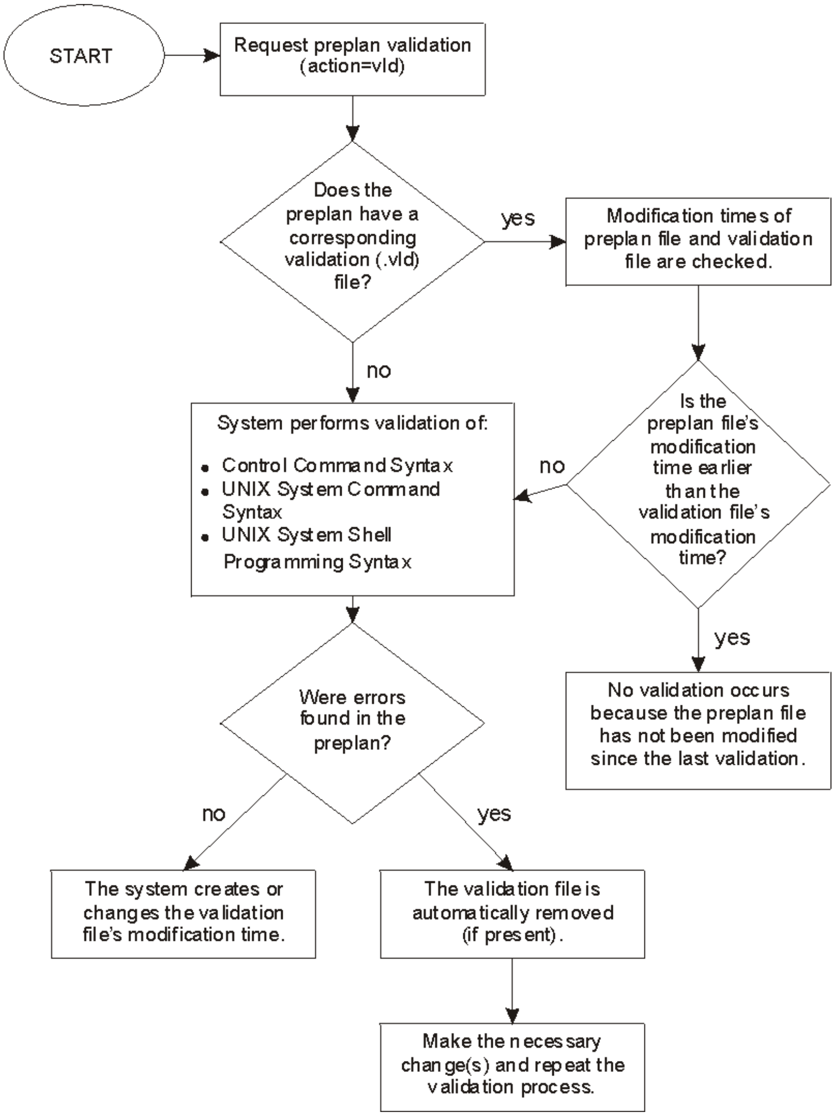
Validating the preplan

After you have created or changed a preplan using the edit command, you must validate it. The validate (`vld`) option checks the preplan (in the order shown) for correctness of:

1. Any control command syntax and parameter values
2. Any *Linux* system command name
3. Its *Linux* system shell programming syntax (in the parameter substitution for the `act` keyword [${1}] and in preplan scripts)

The `vld` option generates a corresponding validation (.vld) file. If there are any errors in validation, you must correct the command syntax where applicable before you can try to validate the preplan again. You cannot use the add or delete (`del`) options until you have successfully validated the preplan. Figure 23 illustrates the validation process.

Figure 23 Preplan validation process


START
Request preplan validation (action=vld)
Does the preplan have a corresponding validation (.vld) file?
yes
Modification times of preplan file and validation file are checked.
no
System performs validation of:
• Control Command Syntax
• UNIX System Command Syntax
• UNIX System Shell Programming Syntax
no
Is the preplan file's modification time earlier than the validation file's modification time?
yes
No validation occurs because the preplan file has not been modified since the last validation.
Were errors found in the preplan?
no
yes
The system creates or changes the validation file's modification time.
The validation file is automatically removed (if present).
Make the necessary change(s) and repeat the validation process.

Adding controls with a preplan

After you have edited and validated a preplan, the preplan is ready to be executed in the network as needed with the add option with the `preplan` command. The system compares the modification times of the preplan and validation files before it executes the preplan.

> **Important!** If you try to use the add option without proper validation, you are prompted for the vld action.

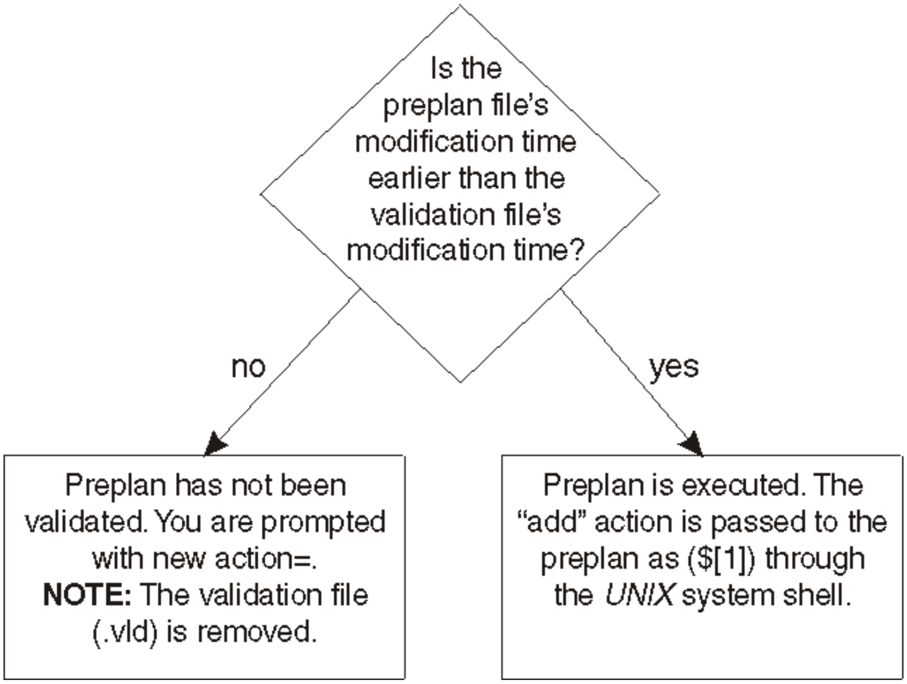
Figure 24 describes the process of adding controls with a preplan.

> **Important!** The name of the preplan is in the *Linux* system shell variable ${0} position, and the preplan action is in the *Linux* system shell variable ${1} position.

Following is an example of adding a preplan.

```
preplan plan1 add
```

Figure 24 Using a preplan to add NTM controls


Is the preplan file's modification time earlier than the validation file's modification time?
no
yes
Preplan has not been validated. You are prompted with new action=. **NOTE:** The validation file (.vld) is removed.
Preplan is executed. The "add" action is passed to the preplan as ($[1]) through the *UNIX* system shell.

Removing controls with a preplan

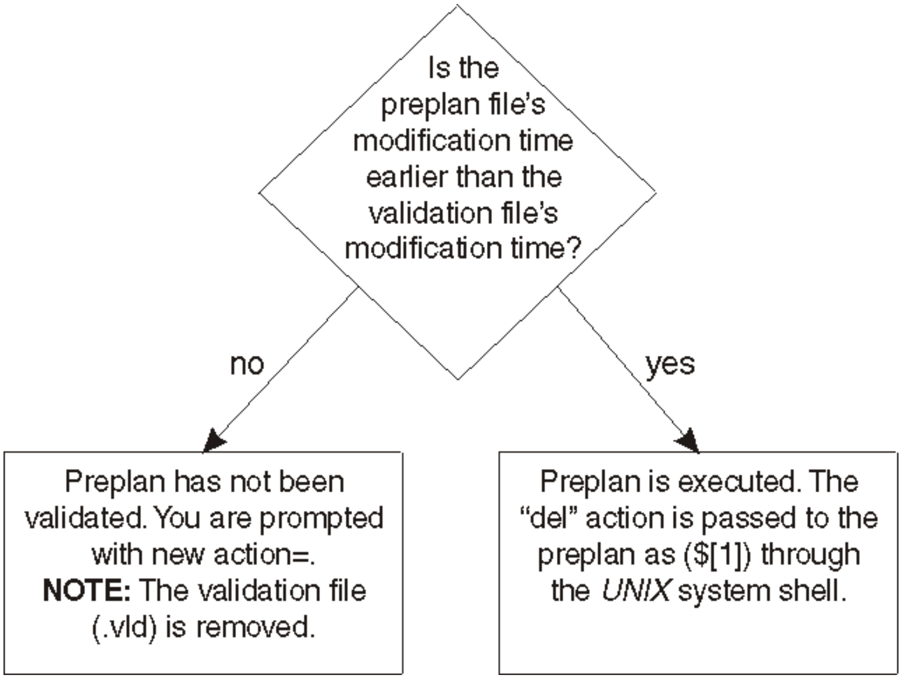
When you want to remove the controls in the preplan from the network, use the `preplan` command with the delete (del) option. The modification times of the preplan and validation files are compared before the controls are removed from the network.

Figure 25 describes the process of removing controls using a preplan.

Following is an example of deleting a preplan.

```
preplan plan1 del
```

Figure 25 Using a preplan to remove NTM controls


Is the preplan file's modification time earlier than the validation file's modification time?
no
yes
Preplan has not been validated. You are prompted with new action=. NOTE: The validation file (.vld) is removed.
Preplan is executed. The "del" action is passed to the preplan as ($[1]) through the UNIX system shell.

Destroying a preplan

To remove a preplan from the "/musr/preplans" directory, use the destroy option with the `preplan` command. If the validation (.vld) file for the preplan is present, it is also removed. When you use the destroy option, you are prompted:

- for the name of the file to be removed
- for a "y" if you want to remove the preplan or "n" if you do not want to remove the preplan

Quitting the preplan

If you do not want to perform any other actions on a preplan, use the quit option. This is the default action when you are prompted for a new action.

Using shell programming in preplans

In addition to creating basic preplans with NTM control commands, you have the option of using shell programming to create preplan scripts. This section provides some examples of preplan scripts created with shell programming.

CAUTION

You should be experienced with shell programming and preplans before attempting to create preplan scripts with shell programming.

Preplan scripts — guidelines

Preplan scripts are basic preplans created with additional *Linux* system shell programming syntax.

Preplan scripts must conform to the same guidelines as those for basic preplans with a few additions.

Follow these additional guidelines to create a basic preplan with shell programming:

- Observe all standard *Linux* system shell programming syntax. Standard *Linux* system shell programming syntax can be verified by execution of “/bin/sh/-nue <*`your program`*>.” Note that this method does not find all errors, and thorough user testing is strongly recommended.
- Make every logical line in the preplan independently executable by the shell. A ***logical*** line is 1 or more lines connected by backslashes (\) at the end of all but the last line. Precede the backslash with a space.
- Use name-defined parameters for each control command used within a preplan. For example, the full {keyword=value(s)} string must be used.
- End each control command with a new line. In cases where, for readability, control commands require more than 1 line, end intermediate lines with a backslash (\). The control command is then treated as 1 continuous line during execution.
- Begin each control command name on a new line within the preplan.
- Use the *Linux* system shell programming parameter substitution for the `act` (action) keyword of each control command. The `preplan` command action values of `add` or `del` (delete) (specified on the `preplan` command line) will be passed as ${1}. No other preplan parameter substitutions are allowed.

Preplan examples

Examples follow to explain script syntax and parameter substitution.

Script syntax

Figure 26 is a correct example of a preplan using script syntax.

Figure 26 Correct script syntax

```
if \
rr from=ofc_A tg=ofc_A-004 act=add \
rrtype=code tfc=alt rout=rate options=htr+cicr \
vias=ofc_C-006+ofc_D-003 vrto=yes+no \
rate=3+10 domains=pots+orig codes=459+614 ;\
then \
  echo "reroute completed for ofc_A" ;\
else \
  echo "reroute failed for ofc_A" ;\
fi
```

Figure 27 is an incorrect example of the same preplan using script syntax. This example is incorrect because the rr command should have been placed on a separate line from the “if” statement.

Figure 27 Incorrect script syntax

```
if rr from=ofc_A tg=ofc_A-004 act=${1} \
rrtype=code tfc=alt rout=rate \
options=htr+cicr \
vias=ofc_C-006+ofc_D-003 vrto=yes+no \
rate=3+10 domains=pots+orig codes=459+614 ;\
then \
  echo "reroute completed for ofc_A" ;\
else \
  echo "reroute failed for ofc_A" ;\
fi
```

Parameter substitution

Figure 28 is a correct example of parameter substitutions in a preplan.

Figure 28 Correct parameter substitution

```
if [ "${1}" = "add" ] ;\
then \
cg office=ofc_B act=add code=459 gap=100 \
annc=eal type=pots tg=ofc_B-004 trap=off \
else \
cg office=ofc_B act=del code=459 gap=100 \
annc=eal type=pots tg=ofc_B-004 trap=off ;\
fi
```

Figure 29 is an incorrect example of parameter substitutions in a preplan. The example is incorrect because the variable ${OFC} is not allowed within preplans.

Figure 29 Incorrect parameter substitution

```
while OFC in `cat ofc_list`
do
total office=${OFC} type=all dintl=dom
done
```

Creating detailed preplans

This section consists of an example (Figure 30) and an explanation (Table 13) of the steps needed to create a detailed preplan. Each line of the example is numbered and some lines are shown with the pound sign (#) symbol. These lines contain comments that guide you through the preplan process.

This example also uses "/bin/ed" as the editor. For example, entering the editor command `1,$p` displays lines 1 through the end of the file.

A detailed explanation of each step is provided in Table 13.

Figure 30 Detailed preplan

```
$ preplan pln1

action = edit
1,$p
   1  #
   2  #  Now you are in the editor for
   3  #  the file /musr/preplans/pln1 until you exit
   4  #  from it.  Remember that lines having a
   5  #  '#' sign in the first column are comments.
   6  #
   7  #  The following  is a sample preplan that
   8  #  will add or delete a re-route.
   9  #
  10  #  MAKE SURE THERE IS ONLY ONE ARGUMENT
  11  #  AND THAT IT IS EITHER 'ADD' OR 'DEL'
  12  #
  13
  14  FAIL=1
  15  OK=0
  16
  17  case ${1} in
  18    "add" )
  19        echo "ADDING RE-ROUTE: PLAN ${0}"
  20        ;;
  21    "del" )
  22        echo "DELETING RE-ROUTE: PLAN ${0}"
  23        ;;
  24    * )
  25        echo "INVALID SYNTAX: ec"
  26        echo "one (and only one) argument ec"
  27        echo "[add or del]"
  28        exit FAIL
  29        ;;
  30  esac
  31
  32  #
```

```
#  BODY OF PLAN
#

if [ "${1}" = "add" ]
then
  if
      cant from=A_ofc tg=B_ofc-004 annc=ea2 \
           alt=25 htr=htr act=add
  then
#
#       SINCE THE CANCEL-TO WORKED,
#       TRY TO DO THE RE-ROUTE
#
      if
          rr from=A_ofc tg=A_ofc-004 \
             rrtype=code tfc=alt rout=rate \
             type=irr options=htr+cicr \
             vias=B_ofc-005+C_ofc-002 \
             vrto=3+10 domains=pots+orig \
             codes=459+614 act=add
      then
          echo "RE-ROUTE PLAN ${0} ec"
          echo "COMPLETED: ec"
          date
      else
#
#           SINCE THE RE-ROUTE DID NOT WORK
#           DELETE THE CANCEL-TO
#
          cant from=A_ofc \
           tg=B_ofc-004 act=del

          exit FAIL
      fi
  else
#
#       THE CANCEL-TO FAILED
#
      echo "CANT FAILED"
      exit FAIL
  fi
else
#
# THE ONLY OTHER CONDITION IS 'DEL' SO,
# DELETE THE CONTROLS IN THE OPPOSITE ORDER
#

  if
      rr from=A_ofc \
 82           to=B_ofc-004 \
 83           rrtype=code act=del
 84
 85
 86    then
 87  #
 88  #     SINCE THE RE-ROUTE WAS SUCCESSFUL
 89  #     DELETE THE CANCEL-TO
 90  #
 91        if
 92            cant from=A_ofc \
 93            tg=B_ofc-004 act=del
 94
 95        then
 96            echo "DELETE RE-ROUTE PLAN ${0}"
 97        else
 98            echo "CONTROLS ONLY PARTIALLY ec"
 99            echo "DELETED: CANT FAILED"
100            exit FAIL
101        fi
102    else
103  #
104  #     DELETE OF RE-ROUTE FAILED
105  #
106        echo "DELETE OF PLAN ${0} FAILED"
107        exit FAIL
108    fi
109  fi
110
111  #
112  # SET THE CALL GAP
113  #
114
115  cg office=ofc_B act=${1} code=459 gap=100 \
116     annc=eal type=pots tg=ofc_B-004 \
117     trap=off
118
119  cg office=ofc_C act=${1} code=459 gap=100 \
120     annc=eal type=pots tg=ofc_B-004 trap=off
121
122  preplan pln2
123
124
125  exit OK
w
2142
q
OK - <pln1>: HAS BEEN CHANGED
<pln1> new action = quit
```

```
IP
$
```

Table 13 Description of Figure 30

Lines	Explanation
1-12	Comment providing a brief description of the preplan.
14-15	Defines exit value variables. (In *Linux* system shell programming, zero (0) is TRUE and non-zero is FALSE). If the preplan exits with a TRUE value, the `preplan` command prints the following error message: `PREPLAN <plan name> WAS UNSUCCESSFUL` Any preplan that does not do a specific “exit” (see lines 28, 65, 72, 100, 107 and 125) exits with the exit value of the last command executed by the preplan.
16-30	A case statement to ensure that the program can only be called with the correct arguments. The `preplan` command passes, as an argument, either `add` or `del`.
36-37	Checks to see if the first argument is `add`. If it is, lines 38-73 are executed. If the argument is not `add`, lines 75-109 are executed.
39-40	A CANT (Cancel-To) control command is executed.
47-52	A REROUTE control command is executed. If it is successful, a message and the date are printed (lines 54-56). If it is not successful, the CANT from line 39 is deleted (lines 62-63).
68-72	The CANT, line 39, is unsuccessful and an error message is printed.
75-109	This section deletes the controls rather than adding them, assuming that the preplan was previously added.
81-85	The REROUTE is deleted. If the deletion is successful, lines 87-101 are then executed. If it is not successful, lines 103-107 are executed.
87-97	The REROUTE deletion is successful, thus the CANT is removed (lines 92-93) and then line 96 is executed. If deletion of the CANT is not successful, lines 98-100 are executed.
98-100	The error message that appears if an error occurs in the CANT deletion.
103-107	The error message that appears if the deletion of the preplan fails.
111-120	2 CALL GAP commands are executed. The first argument to the preplan, `add` or `del`, is substituted for the ${1} before execution.
122	Shows that one preplan may be executed from inside a preplan. The nesting of preplans in this manner should not exceed 2 levels.

□

purglog

Description

The `purglog` (purge log) command removes matched control entries from the database.

1. This command is executed automatically each Saturday morning. You may change the time `purglog` is executed by modifying the cron files.
2. The control log is purged automatically once a week to remove all matched controls that are more than a month old. Before the automatic purge is performed, all matched controls in the control log are saved to the “/musr/log/ctrlog.out” file for future reference.

Syntax

```
purglog type=type [stop=stop]
```

Parameters

`type` Type of controls to purge

- all — remove all matched controls
- cg — remove all matched CG controls
- tg — remove all matched CANT, CANF, SKIP, RR, DPT and CRO controls
- atg — remove all matched ACC, and CR controls
- htr — remove all matched DHTR and IHTR controls

`stop` Stop time for purging controls. Entering the stop time removes all matched entries in the control log that occur before the stop time.

Important! Enter the stop time in the format YY/MM/DD-HH:mm. As a minimum, you must enter the minute. The remaining parameters (current year/month/day/hour) are defaulted. If you do not enter a stop time, all specified matching control types are purged.

Examples

```
$ purglog all
$ purglog type=cg
$ purglog htr 90/04/26-08:00
```

Related command(s)

cron(1) (*Linux* command)
“ctrlog” (p. 59)

□

total

Description

The `total` (total office control) command removes all controls of a particular control type from a particular office(s).

Syntax

```
total  office=office type=type [dom/intl=dom/intl] [mode=mode]
```

Parameters

`office` Office name

`type` Control type

Important! You can enter multiple values separated by the plus sign (+).

Reference: The `total` command only affects the controls listed in the Table 14

`dom/intl` Type of HTR codes (*4ESS* only), where:

- dom = Domestic
- intl = International

`mode` Execution Mode. Valid values are:

- exec = execute the control (default)
- validate = validate the control

Tables

The following table provides more information about the parameters for the `total` command.

Table 14 total control table

Variable	Generic or Switch	Values
type	*4ESS, 5ESS*, 7R/E	canf, cant, skip, rr, cro, cg, all
	GTD-5	canf, cant, skip, orr, irr, str, silc, allrout, dest, term, cg, htr, all
	DMS, LSSGR, SCSN	canf, cant, skip, rr, dptpri, cg, htr, all
	EWSD	canf, cant, skip, rr, cg, all
	1A *ESS*	flex_man, cg, pp, cgx, cgn, all

System responses

Figure 31 shows the responses for the `total` command.

Figure 31 total system responses

IF the command is ...	THEN the system responds with the ...
executed successfully	IP followed by a message (see Figure 32 for an example)
not executed successfully	appropriate error message

Figure 32 shows an example of the type of message displayed when the total command is successfully executed.

Figure 32 Message when the total command executes successfully

```
$ total okld all
Office = okldca0349t, Control type =canf
  IP
Max Waiting time is = 200 sec

Office = okldca0349t, Control type =cant
  IP
Max Waiting time is = 200 sec

Office = okldca0349t, Control type =skip
  IP
Max Waiting time is = 200 sec

Office = okldca0349t, Control type =rr
  IP
Max Waiting time is = 200 sec

Office = okldca0349t, Control type =cro
  IP
Max Waiting time is = 200 sec

Office = okldca0349t, Control type =cg
  IP
Max Waiting time is = 200 sec
```

Examples

```
$ total office=abcd type=all
```

Related command(s)

“ctrlog” (p. 59)

□

5 Record Base and Database Commands

Overview

Purpose

This chapter lists and describes the NTM record base and database commands. Use record base and database commands to add, modify, delete, maintain, and access data in the NTM database.

References

- *Record Base Administration Guide*
- *Data Tables Guide*

Contents

This chapter discusses the following topics:

□

Command functions

Table

Table 1 lists each NTM record base or database command and its function.

Table 1 Record base/database commands and functions (Sheet 1 of 2)

Command	Function
`create`	Creates all or a portion of the record base
`dayend`	Changes current database to next available database
`dbadmin`	Changes status of database
`dbstat`	Provides status of all Tuxedo databases
`dbtest`	Tests "/musr/rb" files for syntax errors and inconsistent data
`demand`	Retrieves specified data from the system database
`fmltoasc`	Converts a fielded buffer to ASCII output
`histdata`	Changes from current mode to histdata mode
`htr_codes`	Lists the codes that belong to an office
`htr_ref`	Lists the offices that belong to a code
`installdb`	Moves a newly built database into the current database area
`limitthr`	Limits the number of exceptions during a crisis.
`longdemand`	Provides the capability to direct typically large processing and response time sql requests to a dedicated database server
`ofc_grep`	Searches "/musr/rb/rspte/rspte" for entries containing office set names
`ofcset`	Retrieves and displays all offices from a given office set
`ongoing`	Retrieves specified data fields from the system's shared memory
`ongoing.test`	Verifies that the SQL (Structured Query Language) has no errors
`pas`	Displays public announcement service data for a specific *4ESS* switch office for the current 5-minute period

Table 1 Record base/database commands and functions (Sheet 2 of 2)

Command	Function
`rbed`	Invokes the record base editor
`rdbstat`	Provides status of all relational databases
`recreate`	Incorporates record base changes into the active database
`tg_grep`	Displays trunk group files within a given trunk group set
`tgset`	Displays trunk groups within a given trunk group set
`thrconvert`	Converts old (static) threshold or filter files into the new rules-based format
`thresh`	Creates a threshold table
`upd_meta`	Change the metadata (labels, aggmode).

□

create

Description

The create command is a record base command. It validates the NTM record base under the “/musr/rb” path (except for the “/musr/rb/dcc_alias” file). The create command also updates the offline database (except for single office create, which updates the current database — not the offline database).

> **Important!** Execution of this command removes the record of all active controls for the TG in the NTM database (tuxedo). All records of active controls present in ORACLE database remain after this option but they have changed a flag (*signed*). Command purglog removes them pernamently from ORACLE.

Another use of the create command is to load data into a standalone reference table that can be used for switch-independent lookups (UDDM types only).

Syntax

```
create files=files  [office=office]

create ref ref_file
```

Before you begin

Run the dbtest command before running the create command. If there are errors, correct the errors until the dbtest command completes with no errors.

Feature restrictions

BDR (Feature 8, “Disaster Recovery (Duplex)” and Feature 40, “Enhanced Disaster Recovery”)

If BDR is installed and active, the record base files will be transferred to and created on all appropriate BDR hosts.

If you want to run create only at your host computer, use the noxfer parameter for global and shared files and the +noxfer parameter for office files.

> **Reference:** See “Differences in record base files” (p. 11) in the *Record Base Administration Guide* for more information on BDR file types.

Examples of the command line entries that would prevent a transfer of the create command to the backup host machine are shown below.

Example: $ `create files=rspte noxfer`

Example: $ `create files=office +noxfer`

Report Writer (Feature 272, "NTM Report Writer")

If report writer is installed and active, the create command will generate files containing entity set, entity reference, trunk group set and trunk group reference data. These files will be sent to the report writer for reporting of the data.

When to use

The create command may be used in four different ways (see Table 2).

Hint: References are in the Record Base Administration Guide.

Table 2 Types of create commands

Options …	The command updates …	Notes / Cautions	See
Full Create			
files=*all*	The offline and temporary databases for all files	**Notes:** 1. The system will not reflect full `create` modifications until the `installdb` command is executed. 2. It is recommended that a full `create` be performed at least once every 30 days. Failure to do so can result in database problems. **CAUTION:** If you have the BDR feature turned on AND run `create all` on both hosts on the same day AND let `recreate` synchronize the databases, THEN the `recreate` may last up to 16 hours. Therefore, it is better to run the `create all` on alternating days or execute the `installdb all` dayend command.	(8-3)
Single File Create			
files=*filename*	The offline and temporary databases for the specified file(s)		(8-6)

Table 2 Types of create commands

Options ...	The command updates ...	Notes / Cautions	See
Single Office Create			

files=*office*
office=*clli*
	All files for a single office that have pathnames ending with the specified office name	**CAUTION:**
		1. Run the `deact` (deactivate) command on an office for all connection types before running the `create` command for a single office.
		2. Do not run the `installdb` command after running the `create` command for a single office. The `create` command for a single office updates the current database and these changes will be lost when you run the `installdb` command.
			(8-9)

Add Data to Standalone Reference Table			
ref *ref_file*	Add data from file to reference table.	**Notes:** 1. The database table must have been created using the command: `manage_uddm add <RefDataType>` 2. If the table already contains reference data, the existing data is removed before data is loaded from the file.	

Parameters

`files` Files to be created:

- all = all record base files
- sets = Sets File only
- alarm = Event_Alarm File
- domain = Domain Acronym File, Office Type Domain File and Control Default Domain File only
- rspte = RSPTE File and TYPXREF File only
- domestic = Domestic Code File only
- intl = International Code File only
- office = Office File, Office Domain File, Transmitter Timeout (TTO) Thresholds File, ATM File and Trunk Group File for a single office (this option updates the current database)
- thresh = all Trunk Group Threshold Files, all ATM Threshold Files, Packet Files, Code Event Files, Mass Call Threshold Files, TG24HourOfl Files, or Job Status Files.

- sched = Threshold Table Schedule File
- pas = PAS Code File only
- discrete = Discrete File for each switch type
- fhc = Final Handling Code (FHC) File only
- inms = INMS File only (BDR only)
- init = Trunk Group File, Office Domain File, Transmitter Timeout (TTO) Thresholds File, and all Trunk Group Threshold Files for a single office (this option updates the current database).

Important! The `init` option does the same thing as the `office` option, except that it is much slower than the normal `create` process. It can be used to resolve database and data collection problems. For example, the init file value will match the order of trunk group suffixes on the Trunk Group Details page to the order in the trunk group file for the office.

- filter = Filter File only
- netevent = processes all files in the *"musr/rb/uddm/<UDDMType>_netevent"* file

`office` Office name. You can enter multiple values separated by the plus sign (+).

`ref_file` Name of the file containing reference data to be loaded into the table of the same name. This file must be in the */musr/rb/ref* directory. See Managing Standalone Reference Tables in the *System Administration Guide* for instructions to create the file.

System responses

Figure 1 shows an example of `create all` command output.

Figure 1 create command output

```
'CREATE ALL' last run: Thu Oct 22 17:50:55 1992
        if you do NOT wish to continue, <del> out
IP
Starting create at Thu Nov 5 13:48:58 1992

Initializing SETS
Creating sets
Initializing DOMAIN
Creating domains
Creating rspte
Creating domestic codes
Creating international codes
Initializing INTENTREF
Initializing ENTSTAT
Initializing AUDITSTAT
Initializing TGREF
Initializing TGIDX
Initializing TGNAMEX
Initializing PTGREF
Initializing PTGNAMEX
Initializing CNIREF
<NOTE:creating office output goes here for each office file>
Creating NTM
Initializing THRTBL
Creating thresh1
Initializing THRSCHD
Initializing THRTIME
Creating sched
Creating PAS codes
Creating ess5 discretes
Creating ess4 discretes
Initializing FHCREF
Creating FHC codes
Creating NTM
Operation backup_dump on database nmdb0:  operation beginning.
Operation backup_dump on database nmdb0:  start of volume on log/bkup
                                             device /dev/rdsk/200s4:
  header Thu Nov 5 14:44:27 1992; sequence 0.
Operation backup_dump on database nmdb0:  section DOMAIN backed up.
Operation backup_dump on database nmdb0:  section SETS backed up.
Operation backup_dump on database nmdb0:  section THRTBL backed up.
Operation backup_dump on database nmdb0:  section THRSCHD backed up.
Operation backup_dump on database nmdb0:  section THRTIME backed up.
Operation backup_dump on database nmdb0:  section INTENTREF backed up.
Operation backup_dump on database nmdb0:  section TGREF backed up.
Operation backup_dump on database nmdb0:  section TGIDX backed up.
Operation backup_dump on database nmdb0:  section TGNAMEX backed up.
Operation backup_dump on database nmdb0:  section PTGREF backed up.
Operation backup_dump on database nmdb0:  section PTGNAMEX backed up.
Operation backup_dump on database nmdb0:  section CNIREF backed up.
Operation backup_dump on database nmdb0:  section FHCREF backed up.
Operation backup_dump on database nmdb0:  section ENTSTAT backed up.
Operation backup_dump on database nmdb0:  section AUDITSTAT backed up.
Operation backup_dump on database nmdb0:  section MTGCTL backed up.
Operation backup_dump on database nmdb0:  section CTL_ENTTG backed up.
Operation backup_dump on database nmdb0:  section ATGCTL backed up.
Operation backup_dump on database nmdb0:  section CTL_ENTATG backed up.
Operation backup_dump on database nmdb0:  section MCGCTL backed up.
Operation backup_dump on database nmdb0:  section CTL_ENTCG backed up.
Operation backup_dump on database nmdb0:  section HTRCTL backed up.
Operation backup_dump on database nmdb0:  section CTL_ENTHTR backed up.
Operation backup_dump on database nmdb0:  section HTRASSN backed up.
Operation backup_dump on database nmdb0:  section PPMTGCTL backed up.
Operation backup_dump on database nmdb0:  section CTL_ENTPP backed up.
Operation backup_dump on database nmdb0:  section HTRTGCTL backed up.
Operation backup_dump on database nmdb0:  section CTL_ENTHTRTG backed up.
Operation backup_dump on database nmdb0:  section HTRPATCTL backed up.
Operation backup_dump on database nmdb0:  section CTL_ENTHTRPM backed up.
Operation backup_dump on database nmdb0:  section NCPUSS backed up.
Operation backup_dump on database nmdb0:  section NCPOVL backed up.
Operation backup_dump on database nmdb0:  section NCPDTO backed up.
Operation backup_dump on database nmdb0:  end of volume on log/bkup
device /dev/rdsk/200s4:
1478 blocks read/written.
Operation backup_dump on database nmdb0:  operation completed normally.
Finished create at Thu Nov 5 14:48:52 1992
Creating dms discretes
Creating ess1 discretes
Creating ess1a discretes
```

Figure 2 shows an example of `create ref` command output.

Figure 2   create ref command output

```
IP
Starting created (pid 12534) at Thu Sep 27 21:17:36 2012

Validating file: '/musr/rb/ref/apc'
Validating file: '/musr/snw/n31/rb/ref/apc'
ud_create successfully completed.
Finished created (pid 12534) at Thu Sep 27 21:17:37 2012

Copying /musr/rb/ref/apc to ntmlnx31, please wait...
Executing /nm/sys/bdr_create on ntmlnx31, please wait...
IP
Validating file: '/musr/rb/ref/apc'
Validating file: '/musr/snw/n41/rb/ref/apc'
ud_create successfully completed.
Starting bdr_create (pid 7533) at Thu Sep 27 21:17:39 2012

Finished bdr_create (pid 7533) at Thu Sep 27 21:17:40 2012
```

Examples

The following example shows a full `create` command for `all` offices:

```
$  create  files=all
```

The following example shows a single file `create` command for the `rspte` file:

```
$  create  files=rspte
```

The following example shows a single office `create` for an `office` file with a nickname “nyrk”:

```
$  create  files=office office=nyrk
```

The following example shows `create ref` for a reference data file named `apc`:

```
$  create  ref  apc
```

References

Chapter 7, “Record Base Administration” and Chapter 8, “Record Base Update Procedures” in the *Record Base Administration Guide*

Related command(s)

“dbtest” (p. 17)
“installdb” (p. 34)
“recreate” (p. 63)
“thresh” (p. 73)

□

□

dayend

Description

The dayend command copies the current database to a historical database. This command does not normally need to be run manually; instead, it is run by the cron daemon overnight. As part of the process, the recreate command is executed automatically — which may stop the system during execution. Use the dbstat command (5-15) to determine available databases.

If the machine is not available, or the dayend command is kept from running, use this command to save the previous day's database.

Syntax

```
dayend
```

Passwords / logins

- If you are logged in as member of snm group other than nmadm , you will be required to enter a password to execute this command.
- If you are logged in as nmadm, no password is required.

Cautions

In order to execute dayend manually, use the sar command to verify that the machine has sufficient system resources (at least 50% idle) to allow the dayend command to complete before the time it is scheduled to run again automatically from cron.

Running the dayend command stops the system if you have scheduled an installdb when=dayend.

The system should not be shut down when the dayend command is running. A lock file “tmp/inhshutdown” will be created when the dayend command is started so that “/etc/shutdown” can determine that dayend is in progress. This lock file will be removed when the dayend command completes successfully.

Related command(s)

“installdb” (p. 34)
“recreate” (p. 63)

□

dbadmin

Description

The `dbadmin` (database administration) command changes the status of any of the historical Tuxedo databases listed in the output of the `dbstat` command.

> **Important!** If there is a problem with a database for example, bad data, follow these rules:
>
> - When the problem is bad enough to warrant wiping out the data currently in the database, clear the database.
> - When the problem can potentially be resolved, mark a database "out of service."

Syntax

```
dbadmin dbname=dbname status=status
```

Parameters

`dbname` Tuxedo historical database name for which the status is to be changed

`status` Status for the given Tuxedo historical database. Choose one of the following:

- active = any database that is available for use
- release = clear and return this database to the historical database rotation
- oos = database is out of service for some reason (bad disk, bad data, etc.)
- hold = take this database out of the historical database rotation
- clear = clear the database; the database will then have no valid data (for example, the start and stop times will be zero)

System responses

There is no system response for this command.

Examples

The following example shows the nmdb2 database being taken out of service:

```
$  dbadmin  dbname=nmdb2 status=oos
```

References

Chapter 8, "Database Administration" in the *System Administration Guide*

See Chapter 8, "Accessing Historical Data" in the *System Overview* for more information on how to use the different databases.

Related command(s)

"dbstat" (p. 15)

□

dbstat

Description

The dbstat (database status) command allows you to view the status of all the tuxedo databases. Possible status values are:

- current
- offline
- hold
- active
- out of service
- needs backed up

Syntax

```
dbstat
```

System responses

Figure 3 illustrates the output displayed on your screen after you run the dbstat command.

Figure 3 dbstat Command Output

```
$ dbstat

          ** Database Status **

STATUS:  offline, active
START: Tue Apr 22 11:43:08 1997     STOP: Tue Apr 22 11:43:08 1997
DBNAME: nmdb0     SRVGRP: 400     DB_INDEX: 0
LOGICAL DEVICES: /dev/ntmdb1/nmdb0, none     DEVICE INDICES: 1, 0

STATUS:  current, active
START: Sun May  4 13:30:00 1997     STOP: Mon May  5 13:30:00 1997
DBNAME: nmdb1     SRVGRP: 500     DB_INDEX: 1
LOGICAL DEVICES: /dev/nmdb1, none     DEVICE INDICES: 2, 0

STATUS:  active, needs backed up
START: Sun May  4 01:05:00 1997     STOP: Mon May  5 01:00:00 1997
DBNAME: nmdb2     SRVGRP: 600     DB_INDEX: 2
LOGICAL DEVICES: /dev/ntmdb1/nmdb3, none     DEVICE INDICES: 3, 0
```

In Figure 3:

- STATUS is the status of the database (current, offline, out of service, active, hold, or needs backed up)
- START is the start time of that database
- STOP is the stop time of that database
- DBNAME is the database Name
- SRVGRP is the Service Group Identifier
- DB_INDEX is the database internal index
- LOGICAL DEVICES are the device names on which the database is created
- DEVICE INDICES are the internal indices used for devices

References

Chapter 8, "Database Administration" in the *System Administration Guide*

See Chapter 8, "Accessing Historical Data" in the *System Overview* for more information on the output shown above.

Related Command(s)

"dbadmin" (p. 13)

□

dbtest

Description

The `dbtest` (database test) command allows you to test the record base files for syntax errors and inconsistent data. It also updates the offline database.

Syntax

```
dbtest files=files [office=office]
```

```
dbtest ref ref_file
```

Before you begin

Run the `dbtest` command before running the `create` command. If there are errors, correct the errors and rerun the `dbtest` command. Repeat this process until `dbtest` does not report any errors.

Feature restrictions

If BDR is installed and active, the record base files will be installed on all appropriate BDR hosts.

If you want to run `dbtest` only at your host computer, use the `noxfer` parameter for global and shared files and the `+noxfer` parameter for office files.

> **Reference:** See "Differences in record base files" (p. 11) in the *Record Base Administration Guide* for more information on BDR file types.

Examples of the command line entries that would prevent a transfer of the `dbtest` command to the backup host machine are shown below.

```
Example: $ dbtest files=rspte noxfer
```

```
Example: $ dbtest files=office +noxfer
```

Parameters

`files` Files to be tested. Choose one:

- all = all record base files
- sets = Sets File only
- alarm = Event_Alarm File
- domain = Domain Acronym File, Office Type Domain File and Control Default Domain File only

- rspte = RSPTE File and TYPXREF File only
- domestic = Domestic Code File only
- intl = International Code File only
- office = Office File, Office Domain File, Transmitter Timeout (TTO) Thresholds File, ATM File, and Trunk Group File for a single office (this option updates the current database)
- thresh = all Trunk Group Threshold Files, all ATM Threshold Files, Packet Files, Code Event Files, Mass Call Threshold Files, TG24HourOfl Files, or Job Status Files
- sched = Threshold Table Schedule File
- pas = PAS Code File only
- discrete = Discrete File for each switch type
- fhc = Final Handling Code (FHC) File only
- inms = INMS File only (BDR only)
- init = Trunk Group File, Office Domain File, Transmitter Timeout (TTO) Thresholds File, and all Trunk Group Threshold Files for a single office (this option updates the current database).

Important! This option does the same thing as the `office` option, except that it is much slower than the normal `create` process. It can be used to resolve database and data collection problems. For example, the init file value will match the order of trunk group suffixes on the Trunk Group Details page to the order in the trunk group file for the office.

- filter = Filter File only

`office` Office name You can enter multiple values separated by the plus sign (+)

`ref_file` Name of the file containing reference data to be loaded into the table of the same name. This file must be in the */musr/rb/ref* directory. See Chapter 16, “UDDM/UDNEI Administration” in the *System Administration Guide* for instructions to create the file.

System responses

Figure 4 shows an example of the `dbtest` command output.

Figure 4 dbtest command output

```
'DBTEST ALL' last run: Thu Oct 22 17:50:55 1992
        if you do NOT wish to continue, <del> out
IP
Starting full dbtest at Thu Nov 5 16:36:50 1992
```

```
Initializing DOMAIN
Initializing SETS
Initializing SETMAP
Initializing THRTBL
Initializing THRSCHD
Initializing THRTIME
Initializing INTENTREF
Initializing ENTSTAT
Initializing AUDITSTAT
Initializing TGREF
Initializing TGIDX
Initializing TGNAMEX
Initializing PTGREF
Initializing PTGNAMEX
Initializing CNIREF
Initializing FHCREF
Creating sets
Creating domains
Creating rspte
Creating sim4e13ent01
  Adding 99 trunk groups to office
  Adding 21 domains to office
Creating cbnmva
Creating db1bed5ent30
Creating dba5e6ent13
  Adding 99 trunk groups to office
  Adding 10 domains to office
Creating thresh1
Creating sched
Creating domestic codes
Creating international codes
Creating PAS codes
Creating ess5 discretes
Creating ess4 discretes
Creating FHC codes
Creating NTM
Finished dbtest at Thu Nov 5 16:57:15 1992
```

Figure 5 shows an example of `dbtest ref` command output for a standalone reference data file named `apc`.

Figure 5 dbtest ref command output

```
IP
Starting dbtestd (pid 17874) at Thu Sep 27 21:37:36 2012

Validating file: '/musr/rb/ref/apc'
Validating file: '/musr/snw/n31/rb/ref/apc'
ud_dbtest successfully completed.
```

```
Finished dbtestd (pid 17874) at Thu Sep 27 21:37:37 2012

Copying /musr/rb/ref/apc to ntmlnx31, please wait...
Executing /nm/cmdbin/dbtestd on ntmlnx31, please wait...
  IP
Validating file: '/musr/rb/ref/apc'
Validating file: '/musr/snw/n41/rb/ref/apc'
ud_dbtest successfully completed.
Starting dbtestd (pid 13170) at Thu Sep 27 21:37:39 2012

Finished dbtestd (pid 13170) at Thu Sep 27 21:37:41 2012
```

Examples

The following example shows a `dbtest` command for all files:

```
$ dbtest files=all
```

The following example shows a `dbtest` command for the RSPTE file:

```
$ dbtest files=rspte
```

The following example shows a `dbtest` command for an office with the nickname of “nyrk”:

```
$ dbtest files=office office=nyrk
```

The following example shows `dbtest ref` for a standalone reference data file named `apc`:

```
$ dbtest ref apc
```

References

Chapter 7, “Record Base Administration” in the *Record Base Administration Guide*

Related command(s)

“create” (p. 5)
“installdb” (p. 34)
“recreate” (p. 63)
“thresh” (p. 73)

□

demand

Description

The `demand` command allows you to retrieve specified data fields from the system database. You specify the field(s) you require and any restrictions upon those fields through an SQL (Structured Query Language) command interpreter.

> **Reference:** See "SQL command syntax" (p. 23) for detailed information on the system SQL interpreter.)

The system SQL commands are read from an external file, *`file.sql`*, the name of which is passed on the command line. The `.sql` extension is not required, but is suggested to facilitate quick and easy identification of system SQL files. If no file name is specified, the `demand` command assumes the SQL commands will be read from standard input (terminated with a **CTRL - D**).

The retrieved data is in binary form. You must pipe or redirect it to some additional command filter/formatter to translate it into a more usable form. The `urwformat` command translates and sorts this binary data into the form required for the *TUXEDO* URW (User Report Writer). The `fmltoasc` command can be used if no *TUXEDO* report is involved.

> **Important!** This command is to be considered deprecated in favor of Oracle database access methods, such as sqlplus, and will be removed in the future. Oracle is the NTM data store and data access via the Tuxedo database is to be discontinued.

Syntax

```
demand  [file=file.sql]
```

Parameters

`file.sql` Name of file containing system SQL commands. If no file is specified, the SQL command interpreter reads commands from standard input until a **CTRL - D** is received.

System responses

The system responds with appropriate error messages if errors are encountered in interpreting the SQL commands. Otherwise, the output appears in binary form conforming to *TUXEDO* FML (Field Manipulation Language).

1. If no data is available in the database, the `demand` command passes a “-1” on to the filter program.
2. Most of the data types are common to both the `ongoing` and `demand` commands, but the field names within those data types may vary for these two commands. If an SQL query for `demand` contains a data type in the FROM clause that is also valid for `ongoing` with field names that are only valid for `ongoing`, `demand` will not respond with error messages. Instead, it ignores the SQL query.

Examples

The following is an example of the `demand` command used in conjunction with the `fmltoasc` command/filter and an SQL file:

```
SELECT to_office suffix period pc ofl %ofl %occ tg_rr_att
   sum_len
FROM tgdat tgctl
WHERE tg_srv = fi and office = grdn and period = 11:55 and
   sum_len = 12
```

Using the `demand` command in conjunction with the `fmltoasc` command/filter, the following *Linux* system command lines:

```
$  demand  file.sql  |  fmltoasc  -pr
$  cat  file.sql  |  demand  |  fmltoasc  -pr
$  demand  <  file.sql  |  fmltoasc  -pr
```

all result in the following conditions for data retrieval:

- Each retrieved record will contain the 9 fields listed in the select clause.
- All of these fields are trunk group data (tgdat) and/or trunk group control (tgctl) data types.
- Only those records that have a trunk group server type of final (fi), have a from office (CLLI code) — in this case the nickname of grdn was used — and have a period of 11:55 (today) or earlier (see below) will be retrieved.
- The last restriction (sum_len = 12) causes each record to be the sum of the data for the last 12 periods ending with the period specified (11:55.)

You can make the same request inside a report source file, using the `urwformat` filter as follows:

```
.exec echo “SELECT to_office suffix period pc ofl %occ
   tg_rr_att sum_len FROM tgdat tgctl WHERE tg_srv = fi and
   office = grdn and period = 11:55 and sum_len = 12” | demand
   | urwformat
```

Important! Although it is shown here on more than one line, the *.exec* line must be on one continuous line within the report source file (in other words, there cannot be any carriage returns).

SQL command syntax

The *TUXEDO* SQL Command Interpreter was designed to be used only with the *TUXEDO* database. Your system has a more efficient database designed to fit the unique type of data being stored and retrieved. You have been supplied with a system SQL command interpreter as part of the `demand` command (hereafter referred to simply as the SQL interpreter) and should ignore references to SQL in the *TUXEDO* documentation.

The SQL command syntax is made up of the following three clauses:

- SELECT `field list`
- FROM `data type list`
- WHERE `condition list`

Important! Only the SELECT and FROM clauses are required.

In the SQL command syntax:

`field list` is a collection of one or more data field names separated by spaces. Use upper- or lower-case letters to specify the field names. Information about database fields can be found in Chapter 1, “All Data Fields” of the *Data Tables Guide* and by using the Search NTM page.

Reference: “HTML search” (p. 4) in the Library Help file.

`data type list`is a collection of one or more data type names separated by spaces. The data type tables are listed in Chapter 2, “Valid Data Types” in the *Data Tables Guide*. The chapter also lists which other data type(s) may be joined with it. Use upper- or lower-case letters to specify the field names.

`condition list`is a collection of one or more data field restrictions to be placed on the data being retrieved. Only those records satisfying the specified condition(s) will be retrieved. The data table lists acceptable restriction values, ranges of values, and/or restriction formats for each field. Use upper- or lower-case letters to specify the field names.

The condition list must be in the format:

```
<restriction> <logical operator> <restriction>
```

where *`<restriction>`* is in the format

```
<field name> <relational operator> <value>
```

In the condition list format:

`field name` is one of the valid field names in Chapter 1, “All Data Fields” of the *Data Tables Guide*

`relational operator`is one of the following valid operators:

- > (greater than)

- >= (greater than or equal to)
- < (less than)
- <= (less than or equal to)
- = (equal to)
- <> (not equal to)

`value` is the value for the comparison, which may be a numeric or a string value. You must enclose string values containing white spaces in quotation marks.

Reference: See the field help files in the online library for a listing of valid field values. Information about database fields can be accessed through the Chapter 1, "All Data Fields" in the *Data Tables Guide* and by using the Search NTM page (see "HTML search" (p. 4) in the Library Help file).

`logical operator`is one of the following:

- and - the following <condition> is and'ed with the previous (and joins two unlike data fields such as n2w > 0 and n1wo > 0, or two like data fields such as ofl > 10 and ofl <= 100).
- or - the following <condition> is or'ed with the previous (or joins two like data fields such as office=sm01 or office=barcln5x).

CAUTION

Exercise care when using the and'ing/or'ing of restrictions. Restrictions such as "WHERE office = sm01 and office = sm02" will result in no records being retrieved. It should also be noted that wide encompassing restriction, or the lack of any restrictions may result in the retrieval of hundreds of thousands of records and an extremely long execution time.

References

See the *TUXEDO FML Programmer's Guide* for detailed information on FML.

Related command(s)

"fmltoasc" (p. 25)
"ongoing" (p. 51)
"ongoing.test" (p. 55)
"start_arc" (p. 19)
"urwformat" (p. 31)

□

fmltoasc

Description

The fmltoasc (FML to ASCII) command converts a fielded buffer to ASCII output according to the various specified options. The Field Manipulation Language (FML) is a set of C-language functions for defining and manipulating data storage structures called ***fielded buffers***. Each fielded buffer contains attribute-value pairs called ***fields***. The attributes are the field names and the values represent their data content.

This command reads its input from standard input (your keyboard) and writes the output to standard output (your terminal screen).

It is intended for use in conjunction with the demand and ongoing commands. The demand command is used to retrieve historical data from the system database, while the ongoing command is used to retrieve certain data from shared memory on a continuous basis. The output of the demand and ongoing commands is in the form of FML fielded buffers, which is binary data. The fmltoasc command converts that binary data into ASCII format. This command reads its input from standard input (your keyboard) and writes the output to standard output (your terminal screen).

> **Important!** This command is to be considered deprecated in favor of Oracle database access methods, such as sqlplus, and will be removed in the future. Oracle is the NTM data store and data access via the Tuxedo database is to be discontinued.

Syntax

```
fmltoasc [-print option] [-f <strg>] [-m <strg>] [-s <char>] [-w
   <char>] [-n <char>] [-c <char>] [-i <strg>] [-t <n>] [-E]
```

Parameters

This command takes the form of a standard *Linux* command with the following options:

print option Valid values are:

- p = Print the data in the form <fieldname>=<value>.
- r = Print only the retrieved fields, that is, those from the select clause of the SQL file. The default is to print all the fields contained within the fielded buffer.

  Reference: "SQL command syntax" (p. 23)
- e = Print the value and exception level for calculated fields. The default is to print only the value.
- d = Use a predefined output format suitable for the DCS (Display Construction Set).
- h = Print output data in a column format with one header. The header will be printed, even if there isn't any data to return for the particular SQL.

- o = Print all the information held in the calc field in 5 columns:
  - print value — actual value of field
  - suspect — 0 = raw count not available; 8 = failure of raw count retrieval; 12 = couldn’t perform indicated calculation properly; 14 = data suspect; 15 = data valid
  - incmplt — 0 = no, complete; 1 = yes, incomplete
  - x10 (times 10) — 0 = no, not times 10; 1 = yes, times 10
  - threshold level — actual threshold level for the field
- l = Print the label associated with the select clause. The label is optional in the SQL file, therefore this option may display a null string.

`-f <strg>` Use the string <strg> as the field delimiter string. The default delimiter string is the blank.

Important! Do not use the `-f` option with the `-w` and/or `-n` options because the `-f` option takes precedence.

`-m <strg>` Use the string <strg> as the message delimiter string. The default delimiter string is a new line.

`-s <char>` Use the character <char> as the subfield delimiter character. The default delimiter character is a blank. This option is used when a field is made up of several subfields.

`-w <char>` Use the character <char> as the delimiter character before a string. The default delimiter character is a blank. This option is used mainly for formatting messages for the Display Construction Set User Interface Management System.

Important! Do not use the `-w` option with the `-f` option because the `-f` option takes precedence.

`-n <char>` Use the character <char> as the delimiter character before a number. The default delimiter character is a blank.

Important! Do not use the `-n` option with the `-f` option because the `-f` option takes precedence.

`-c <char>` Use the character <char> as the delimiter between fields in calculated fields. The default delimiter character is a blank.

`-i <strg>` Use the string <strg> as the invalid data string. The default invalid data string is “-1”.

`-t <n>` Use the number <n> as the time format definition, where:

- n = 1 *UNIX* date format
- n = 2 hour and minute

- n = 3 date, hour and minute
- n = 4 hour, minute, and second (default)
- n = 5 date, hour, minute, and second

`-E` Expand each discrete to include its DISCRETE_NAME, LABEL, and LEVEL, along with its current value.

This new -E option will return a -1 for the LABEL and LEVEL values in the following two cases:

1. The discrete is not valid for the office type of the selected record. The discrete will have a value of -2 in this case.
2. The select line did not request either OFFICE or TYPE. One of these two fields is necessary for `fmltoasc` to know the office type of the selected record so it can know which discrete file contains the LABEL and LEVEL.

System responses

The converted and formatted data is written to standard output.

Examples

The following command line shows the `fmltoasc` command used with the `demand` command.

```
$ demand file.sql | fmltoasc -pr -f ":"
```

Figure 6 shows the contents of the SQL file (file.sql) on the `fmltoasc` and `demand` command line.

Figure 6 SQL File Used with fmltoasc and demand Commands

```
SELECT to_office pc %ofl
FROM   tgdat
WHERE  tg_svr = fi and office = kscymo0905t
```

This command formats the retrieved fields, fields specified in the FROM select clause of the SQL file, and fields found in the FML fielded buffer output of the `demand` command in the following format:

```
TO_OFFICE=KSCYMO0905T:PC=1234:%OFL=12
```

The output will include peg count and percentage of overflow only for final trunk groups from the kscymo0905t office. The normal field delimiter string (space) is replaced with a colon (:).

References

See the `demand` command (5-21) for a list of the retrievable fields for each command and an explanation of the SQL.

Chapter 3, “SQL Interpreter” in the *Data Tables Guide*

See the *TUXEDO FML Programmer's Guide* for detailed information on FML.

Related command(s)

“demand” (p. 21)
“ongoing” (p. 51)
“ongoing.test” (p. 55)
“urwformat” (p. 31)

□

histdata

Description

The “Historical Data Across Releases” (HISTDATA) feature offers network managers the ability to access historical database information from two previous generics of NTM

The `histdata` command changes the current mode to HISTDATA mode. The system prompt changes to “HISTDATA>”. The time period (for data viewing purposes) changes to the time entered as an option to the `histdata` command. The time period entered should be that of data collected under a previous generic.

Syntax

```
histdata time
```

Feature restrictions

This command is available only with the purchase of Feature 123, “Historical Data Across Releases”.

Before you begin

Before running `histdata`, the **SAM** option needs to be run to read in the archive data stored under a previously loaded generic.

Parameters

`time` The time you want to look at the appropriate database. Enter the local time in the form: [[[[YY/]MM/]DD]-hh:]mm, where:

- YY = year (last 2 digits of the year)
- MM = month (01 through 12)
- DD = day of the month (01 through 31)
- hh = hour (00 through 23)
- mm = minute (00 through 59)

System responses

Figure 7 shows sample output of the `histdata` command.

Figure 7 histdata Output

```
$ histdata 94/02/10-12:00
Your environment has been modified to enable the execution of software
from the 3.0net generic.  Only the commands listed in the user
   documentation
```

```
as being valid for this feature will be available while in this mode.
   Your
environment will be returned to its original settings when you run the
current command.

HISTDATA>
```

References

See Chapter 8, “Accessing Historical Data” in the *System Overview* for more information on Feature 123, “Historical Data Across Releases”.

See Chapter 5, “Backing Up and Restoring the System” in the *System Administration Guide* for more information on using SAM.

Related command(s)

“dbadmin” (p. 13)
“dbstat” (p. 15)

□

htr_codes

Description

The htr_codes command searches for an office or country within the record base files (intl or domestic) and returns a list of codes that belong to that office or country.

Syntax

```
htr_codes
```

Feature restrictions

With BDR installed, the htr_codes command will search through all the record base partitions when necessary.

Parameters

The htr_codes command does not take arguments on the command line. The command will prompt you for the file type, which can be intl (i) or domestic (d), and the country name or office. If your input is i, you will be asked to enter a country name. If your input is d, you will be asked to enter an office name.

System responses

Figure 8 shows the responses for the htr_codes command.

Figure 8 htr_codes system responses

IF the command is …	THEN the system responds with …
executed successfully	IP Enter INTL (i) or DOMESTIC (d): d Enter an office name: clmboh0101tx

Figure 9 shows an example of the output of the htr_codes command when the command is executed as shown in Figure 8.

Figure 9 htr_codes command output

```
------------Codes belonging to clmb0h0101tx--------------

001         002         00354       004
005         006502      007629      008
009221      237

---- A total of 10 codes were found for clmboh0101tx -----
```

Related command(s)

"htr_ref" (p. 33)

□

htr_ref

Description

The htr_ref command searches for a domestic code or a country code within the record base files (intl or domestic) and returns the office to which that code belongs.

Syntax

```
htr_ref
```

Feature restrictions

With BDR installed, the htr_ref command will search through all the record base partitions when necessary.

Parameters

The htr_ref command does not take arguments on the command line. The command will prompt you for the file type, which can be intl (i) or domestic (d). You will then be prompted to enter a code.

System responses

Figure 10 shows the responses for the htr_ref command.

Figure 10 htr_ref system responses

IF the command is ...	THEN the system responds with ...
executed successfully	IP Enter INTL (i) or DOMESTIC (d): d Enter a CODE: 237

Figure 11 shows an example of the output of the htr_ref command when the command is executed as shown in Figure 10.

Figure 11 htr_ref command output

```
Code = 237          Reference Office = clmboh0101tx
```

Related command(s)

"htr_codes" (p. 31)

□

installdb

Description

The installdb (install database) command is a record base command used to move a newly built database (residing in the offline area) into the current database area.

This database is either moved immediately or when the dayend command runs, depending on the value you specify for the when parameter.

Syntax

```
installdb  files=files when=when
```

Passwords / logins

You must be nmadm to run this command.

Before you begin

You do not need to run the installdb command after running a single office create. The current database will already be updated.

Cautions

1. If you select the "now" option for the when parameter, you must stop NTM before running this command.

   Reference: Chapter 4, "Starting and Stopping the System" in the *System Administration Guide*

2. To avoid losing any database changes, it is recommended that you execute a full install (installdb files=all when=now) immediately after a full create (create files=all) is performed. If you choose to install the files at dayend, any reference data changes made between the full create and dayend may be lost. Also, if the create files=inms command is run between the full create and dayend, changes to all but the inms file will be lost.
3. Error messages received when when=dayend are mailed to the user "nmadm".

Feature restrictions

If BDR is installed and active, the record base files will be installed on all appropriate BDR hosts.

If you have purchased Feature 41, "Install RSPTE Without Stopsys" you do not need to stop the system before running the `installdb rspte` command. Changes to the currently defined offices will go into effect immediately while new RSPTE entries will not be recognized by the system servers until the servers are restarted with the `stopsys` and `startsys` commands.

Parameters

`files` Files to be installed. Choose one:

- all = all files

  Important! This option removes the record of all active controls for the current day for the NTM database (tuxedo). All records of active controls present in ORACLE database remain after this option but they have changed a flag (signed). Command `purglog` removes them pernamently from ORACLE.
- reference = reference data only (this does NOT remove the records of any controls)
- rspte = RSPTE File only
- sets = Sets File only
- domain = Domain Acronym File and Control Default Domain File only
- thresh = all Trunk Group Threshold Files, all ATM Threshold Files, Packet Files, Code Event Files, Mass Call Threshold Files, TG24HourOfl Files, or Job Status Files
- sched = Threshold Table Schedule File
- domestic = Domestic Code File only
- intl = International Code File only
- pas = PAS Code File only
- discrete = Discrete File for all switch types
- fhc = Final Handling Code (FHC) File only
- inms = INMS File (BDR only)
- filter = Filter File only
- alarm = Event_Alarm File

`when` When to install the database. Choose one:

- now = perform installation immediately
- dayend = perform installation as part of dayend automatic routines
- cancel = cancel a previously scheduled installdb

  Important! If you select `when=now`, you must stop NTM before running this command.

Reference: Chapter 4, “Starting and Stopping the System” in the *System Administration Guide*

System responses

Figure 12 shows the responses for the `installdb` command.

Figure 12 installdb system responses

IF the command is ...	THEN the system responds with ...
executed successfully	IP followed by the output shown in Figure 13.
not executed successfully	the appropriate error message.

Figure 13 shows the output of the command `installdb all now`.

Figure 13 installdb Command Output

```
$ installdb all now

INSTALLDB ALL last run: Thu Nov 5 10:09:30 1992
        if you do NOT wish to continue, <del> out
  IP
Operation backup_load on database nmdb1:  operation beginning.
Operation backup_load on database nmdb1:  start of volume on log/bkup device
/dev/rdsk/sc1d5s9:
  header Thu Nov 5  15:17:18 1992; sequence 0.
Operation backup_load on database nmdb1:  section DOMAIN recovered from backup.
Operation backup_load on database nmdb1:  section SETS recovered from backup.
Operation backup_load on database nmdb1:  section THRTBL recovered from backup.
Operation backup_load on database nmdb1:  section THRSCHD recovered from backup.
Operation backup_load on database nmdb1:  section THRTIME recovered from backup.
Operation backup_load on database nmdb1:  section INTENTREF recovered from backup.
Operation backup_load on database nmdb1:  section TGREF recovered from backup.
Operation backup_load on database nmdb1:  section TGIDX recovered from backup.
Operation backup_load on database nmdb1:  section TGNAMEX recovered from backup.
Operation backup_load on database nmdb1:  section PTGREF recovered from backup.
Operation backup_load on database nmdb1:  section PTGNAMEX recovered from backup.
Operation backup_load on database nmdb1:  section CNIREF recovered from backup.
Operation backup_load on database nmdb1:  section FHCREF recovered from backup.
Operation backup_load on database nmdb1:  section ENTSTAT recovered from backup.
Operation backup_load on database nmdb1:  section AUDITSTAT recovered from backup.
Operation backup_load on database nmdb1:  section ATGCTL recovered from backup.
Operation backup_load on database nmdb1:  section CTL_ENTATG recovered from backup.
Operation backup_load on database nmdb1:  section MCGCTL recovered from backup.
Operation backup_load on database nmdb1:  section CTL_ENTCG recovered from backup.
Operation backup_load on database nmdb1:  section HTRCTL recovered from backup.
Operation backup_load on database nmdb1:  section CTL_ENTHTR recovered from backup.
Operation backup_load on database nmdb1:  section HTRASSN recovered from backup.
Operation backup_load on database nmdb1:  section PPMTGCTL recovered from backup.
Operation backup_load on database nmdb1:  section CTL_ENTPP recovered from backup.
Operation backup_load on database nmdb1:  end of volume on log/bkup device
/dev/rdsk/sc1d5s9:
  416 blocks read/written.
Operation backup_load on database nmdb1:  operation completed normally.
```

Examples

The following example schedules an `installdb` command to be run during the `dayend` procedures:

```
$  installdb  files=all when=dayend
```

References

"Performing a full create and installdb" (p. 3) in the *Record Base Administration Guide*

Related command(s)

"create" (p. 5)
"recreate" (p. 63)
"startsys" (p. 23)
"stopsys" (p. 28)

□

limitthr

Description

Use the `limitthr` command to control the automatic limiting of exception records displayed on the alert tables, maps, and `ongoing`. Limited Presentation Mode is activated when the number of exception records exceeds the system-wide heavy load upper limit. For each data type, the lower level exception records are not made available to `ongoing`, limiting the exception records to the system-wide limit for that data type. The `limitthr` command allows the system-wide limits to be lowered to 75%, 66%, or 50%. Limited Presentation Mode can also be disabled at the risk of degrading system performance during exception overload.

Syntax

```
limitthr action
```

Parameters

`action` Valid values include:

- `100` — sets the system-wide exception limit to the “Heavy Load Upper Limit” value.
- `75` — reduces the system-wide exception limit to 75% of the “Heavy Load Upper Limit” value.
- `66` — reduces the system-wide exception limit to 66% of the “Heavy Load Upper Limit” value.
- `50` — reduces the system-wide exception limit to 50% of the “Heavy Load Upper Limit” value.
- `stat` — displays the status of the Limited Presentation Mode
- `disable` — removes the system-wide limit of exception records, effectively turning off the “Limited Presentation Mode”

WARNING

Selecting the “`disable`” option disables the system’s automatic limiting of the number of exception records that can be displayed. Choosing this option may have serious performance implications. During a crisis, forcing the system to attempt to display all exception records will consume a tremendous amount of system resources. The ability to generate reports may be lost. The system may become so overloaded that the data collection function may

cease functioning and the time to respond to user control inputs may become excessive.

Exception record limit tables

In the following tables, 100 is the heavy load upper limit. Limitations vary according to the NTM configuration installed.

Table 3 **Exception record limits with GTD-5 switches (Small/Medium NTM configuration)**

With GTD-5 Switches	**100**	**75**	**66**	**50**
System-Wide Limit	2000	1500	1320	1000
Hysteresis Limit	1800	1350	1188	900
Data Type				
TG-TGDAT	1692	1269	1116	846
OFC-ENTDAT	175	131	116	88
CIC/FG-TTODAT	27	20	18	13
LINK-HRLKDAT	66	50	44	33
UNIT-PUPDAT	40	30	26	20

Table 4 **Exception record limits without GTD-5 switches (Small/Medium NTM configuration)**

Without GTD-5 Switches	**100**	**75**	**66**	**50**
System-Wide Limit	2000	1500	1320	1000
Hysteresis Limit	1800	1350	1188	900
Data Type				
TG-TGDAT	1753	1315	1157	876
OFC-ENTDAT	215	161	142	108
CIC/FG-TTODAT	32	24	21	16

Table 5 **Exception record limits with GTD-5 switches (Large NTM configuration)**

With GTD-5 Switches	**100**	**75**	**66**	**50**
System-Wide Limit	4000	3000	2640	2000
Hysteresis Limit	3600	2700	2376	1800

Table 5 **Exception record limits with GTD-5 switches (Large NTM configuration)**

With GTD-5 Switches	100	75	66	50
Data Type				
TG-TGDAT	3384	2538	2232	1692
OFC-ENTDAT	350	262	232	176
CIC/FG-TTODAT	54	40	36	26
LINK-HRLKDAT	132	100	88	66
UNIT-PUPDAT	80	60	52	40

Table 6 **Exception record limits without GTD-5 switches (Large NTM configuration)**

Without GTD-5 Switches	100	75	66	50
System-Wide Limit	4000	3000	2640	2000
Hysteresis Limit	3600	2700	2376	1800
Data Type				
TG-TGDAT	3506	2630	2314	1752
OFC-ENTDAT	430	322	284	216
CIC/FG-TTODAT	64	48	42	32

System responses

When you enter the `limitthr` command, the system returns a response showing the settings as selected in the command syntax. This response shows the threshold values for a percentage of the heavy load upper limit settings. See “Exceptions” (p. 8) in the *System Overview* for heavy load limits.

Examples

Example 1

```
limitthr act=stat
IP
limitthr: Limited Presentation Mode is disabled
```

Example 2

```
limitthr 66
IP
limitthr: Current Exception threshold limit is set at 66%
```

```
limitthr: limiting is ON at 1320 total exceptions, OFF at 1188 exceptions
limitthr:      TG-TGDAT limit is 1156 exceptions
limitthr:      OFC-ENTDAT limit is 141 exceptions
limitthr:      CIC/FG-TTODAT limit is 21 exceptions
limitthr: Limited Presentation Mode is OFF
```

Example 3

```
$limitthr 50
IP
limitthr: Current Exception threshold limit is set at 50%
limitthr: limiting is ON at 1000 total exceptions, OFF at 900 exceptions
limitthr:      TG-TGDAT limit is 876 exceptions
limitthr:      OFC-ENTDAT limit is 107 exceptions
limitthr:      CIC/FG-TTODAT limit is 16 exceptions
limitthr: Limited Presentation Mode is ON
```

Example 4

```
limitthr disable
IP
limitthr: Limited Presentation Mode is disabled
```

References

"Inhibiting trunk group thresholds" (p. 6) in the *System Overview*

Related command(s)

"ongoing" (p. 51)

□

longdemand

Description

The longdemand command provides the capability to direct typically large processing and response time sql requests to a dedicated database server. This allows long report requests to complete without timing out and avoids congesting the normal database servers used for demand requests.

Important! This command is to be considered deprecated in favor of Oracle database access methods, such as sqlplus, and will be removed in the future. Oracle is the NTM data store and data access via the Tuxedo database is to be discontinued.

Syntax

```
longdemand  [file=file.sql]
```

Restrictions

The longdemand command operates on the current/online database only; historical database access is not supported.

Parameters

file.sql — You can place your system SQL commands in a file to be read by the longdemand command. If no file is specified, the SQL command interpreter reads commands from standard input until a **CTRL - D** is received.

Detailed description

The longdemand command is the same as demand except that it sets time out intervals of 3 hours rather than the 5 minutes for database requests and responses. Longdemand is intended to serve sql requests that are not required to provide real-time information, but to support report requests, typically spawned through using the cron facility, that can tolerate possibly long delays in response time.

The longdemand command allows you to retrieve specified data fields from the system database. You specify the field(s) you require and any restrictions upon those fields through an SQL (Structured Query Language) command interpreter.

Reference: "SQL command syntax" (p. 23)

The system SQL commands are read from an external file, `file.sql`, the name of which is passed on the command line. The `.sql` tag is not required, but is suggested to facilitate quick and easy identification of system SQL files. If no file name is specified, the `longdemand` command assumes the SQL commands will be read from standard input (terminated with a **CTRL - D**).

The retrieved data is in binary form. You must pipe or redirect it to some additional command filter/formatter to translate it into a more usable form. The `urwformat` command translates and sorts this binary data into the form required for the *TUXEDO* URW (User Report Writer). The `fmltoasc` command can be used if no *TUXEDO* report is involved.

System responses

The system responds with appropriate error messages if errors are encountered in interpreting the SQL commands. Otherwise, the output appears in binary form conforming to *TUXEDO* FML (Field Manipulation Language).

1. If no data is available in the database, the `longdemand` command passes a “-1” on to the filter program.
2. Most of the data types are common to both the `ongoing` and `longdemand` commands, but the field names within those data types may vary for these two commands. If an SQL query for `longdemand` contains a data type in the FROM clause that is also valid for `ongoing` with field names that are only valid for `ongoing`, `longdemand` will not respond with error messages. Instead, it ignores the SQL query.

Examples

The following is an example of the `longdemand` command used in conjunction with the `fmltoasc` command/filter and an SQL file:

```
SELECT     to_office suffix period pc ofl %ofl %occ tg_rr_att
   sum_len
FROM      tgdat tgctl
WHERE      tg_srv = fi and office = grdn and period = 11:55 and
   sum_len = 12
```

Using the `longdemand` command in conjunction with the `fmltoasc` command/filter, the following *Linux* system command lines:

```
$  longdemand  file.sql  |  fmltoasc  -pr
$  cat  file.sql  |  longdemand  |  fmltoasc  -pr
$  longdemand  <  file.sql  |  fmltoasc  -pr
```

all result in the following conditions for data retrieval:

- Each retrieved record will contain the 9 fields listed in the select clause.
- All of these fields are trunk group reference (tgdat) and/or trunk group control (tgctl) data types.

- Only those records that have a trunk group server type of final (fi), have a from office (CLLI code) — in this case the nickname of grdn was used — and have a period of 11:55 (today) or earlier (see below) will be retrieved.
- The last restriction (sum_len = 12) causes each record to be the sum of the data for the last 12 periods ending with the period specified (11:55.)

You can make the same request inside a report source file, using the `urwformat` filter as follows:

```
.exec echo “SELECT to_office suffix period pc ofl %occ
  tg_rr_att sum_len FROM tgdat tgctl WHERE tg_srv = fi and
  office = grdn and period = 11:55 and sum_len = 12” |
  longdemand | urwformat
```

Important! Although it is shown here on more than one line, the *.exec* line must be on one continuous line within the report source file (in other words, there cannot be any carriage returns).

SQL command syntax

The *TUXEDO* SQL Command Interpreter was designed to be used only with the *TUXEDO* database. Your system has a more efficient database designed to fit the unique type of data being stored and retrieved. You have been supplied with a system SQL command interpreter as part of the `longdemand` command (hereafter referred to simply as the SQL interpreter) and should ignore references to SQL in the *TUXEDO* documentation.

The SQL command syntax is made up of the following three clauses:

- SELECT *`field list`*
- FROM *`data type list`*
- WHERE *`condition list`*

Notes: Only the SELECT and FROM clauses are required.

`In the SQL command syntax:`

`field list` is a collection of one or more data field names separated by spaces. Use upper- or lower-case letters to specify the field names. Information about database fields can be found in Chapter 1, “All Data Fields” of the *Data Tables Guide* and by using the Search NTM page.

Reference: “HTML search” (p. 4) in the Library Help file.

`data type list`is a collection of one or more data type names separated by spaces. The data type tables are listed in Chapter 2, “Valid Data Types” in the *Data Tables Guide*. The chapter also lists which other data type(s) may be joined with it. Use upper- or lower-case letters to specify the field names.

`condition list`is a collection of one or more data field restrictions to be placed on the data being retrieved. Only those records satisfying the specified condition(s) will be retrieved. The data table lists acceptable restriction values, ranges of values, and/or restriction formats for each field. Use upper- or lower-case letters to specify the field names.

The *`condition list`* must be in the format:

`<restriction> `*`logical operator`*` <restriction>`

where *`<restriction>`* is in the format

`field name relational operator value`

In the *`<restriction>`* format:

`field name` is one of the valid field names in Chapter 1, "All Data Fields" of the *Data Tables Guide*

`relational operator`is one of the following valid operators:

- > (greater than)
- >= (greater than or equal to)
- < (less than)
- <= (less than or equal to)
- = (equal to)
- <> (not equal to)

`value` is the value for the comparison, which may be a numeric or a string value. You must enclose string values containing white spaces in quotation marks.

Reference: See the field help files in the online library for a listing of valid field values. Information about database fields can be accessed through the Chapter 1, "All Data Fields" in the *Data Tables Guide* and by using the Search NTM page (see "HTML search" (p. 4) in the Library Help file).

`logical operator`is one of the following:

- and — the following <condition> is and'ed with the previous (and joins two unlike data fields such as n2w > 0 and n1wo > 0, or two like data fields such as ofl > 10 and ofl <= 100).
- or — the following <condition> is or'ed with the previous (or joins two like data fields such as office=sm01 or office=barcln5x).

CAUTION

Exercise care when using the and'ing/or'ing of restrictions. Restrictions such as "WHERE office = sm01 and office = sm02" will

result in no records being retrieved. It should also be noted that wide encompassing restriction, or the lack of any restrictions, may result in the retrieval of hundreds of thousands of records and an extremely long execution time.

References

See the *TUXEDO FML Programmer's Guide* for detailed information on FML.

Related command(s)

"fmltoasc" (p. 25)
"ongoing" (p. 51)
"ongoing.test" (p. 55)
"start_arc" (p. 19)
"urwformat" (p. 31)

□

ofc_grep

Description

The ofc_grep (duplicate office set check) command searches the RSPTE File in the record base to see if an office set name exists.

Syntax

```
ofc_grep officeset
```

Feature restrictions

With BDR installed, the ofc_grep command also searches through all the record base partitions.

Parameters

officeset	Name of the desired office set

System responses

Figure 14 illustrates the output for the ofc_grep command.

Figure 14 ofc_grep command output

```
$ ofc_grep eadn

The set eadn does not exist for external offices in the /musr/rb/rspte/rspte file.

The set eadn does not exist for internal offices in the /musr/rb/rspte/rspte file.

The set eadn does not exist for external offices in the /musr/snw/pbso/rb/rspte/rspte
   file.

The set eadn already exists for internal offices in the /musr/snw/pbso/rb/rspte/rspte
   file.

The set eadn does not exist for external offices in the /musr/snw/pbno/rb/rspte/rspte
   file.

The set eadn already exists for internal offices in the /musr/snw/pbno/rb/rspte/rspte
   file.

The set eadn does not exist for external offices in the /musr/snw/ind/rb/rspte/rspte
   file.

The set eadn already exists for internal offices in the /musr/snw/ind/rb/rspte/rspte
   file.
```

```
The set eadn does not exist for external offices in the /musr/snw/gtd5/rb/rspte/rspte
   file.

The set eadn already exists for internal offices in the /musr/snw/gtd5/rb/rspte/rspte
   file.

The set eadn already exists in the /musr/rb/sets/sets file
$
```

Examples

Use the following command line to generate the output shown in Figure 14:

```
$  ofc_grep
```

Or to generate the same output without waiting for the system to prompt you, you can enter the command as follows:

```
$  ofc_grep  eadn
```

Related command(s)

“tg_grep” (p. 67)

□

ofcset

Description

The ofcset (office set check) command retrieves and displays all offices in a given office set. To ensure that the set name you enter is valid, check it first in the "/musr/rb/sets/sets" file.

Syntax

```
ofcset [set=officeset]
```

Feature restrictions

With BDR installed, the ofcset command also searches through all the record base partitions.

Restrictions

This command cannot be used by more than one person at a time. If you attempt to execute this command when it is already in use, you will receive a message asking you to try again later. Please ensure that no one else is running this command before trying again.

Parameters

officeset	Name of the desired office set

System responses

Figure 15 illustrates the output for the ofcset command.

Figure 15 ofcset Command Output

```
$ ofcset
setname=dm28
Begin ofcset at: Wed Jun21 14:40:06 EDT 1995

  IP
The offices in set dm28 are:

OFFICE
------
bakrca11ds0
enctca12ds0
fntaca11ds0
frckca11ds0

End ofcset at:  Wed Jun 21 14:40:07 EDT 1995
```

```
$
```

Examples

To generate the output above without waiting for the system to prompt you, you can enter the command as follows:

```
$ ofcset dm28
```

Related command(s)

“tgset” (p. 69)

□

ongoing

Description

The ongoing command continuously retrieves specified data fields from the system's shared memory. The field(s) that you require and any restrictions upon those fields are specified through an SQL (Structured Query Language) command interpreter. The system SQL commands are read from an external file, file.sql, the name of which is passed on the command line.

> **Important!** The .sql tag is not required, but is suggested to facilitate identification of system SQL files. If no file name is specified, the ongoing command assumes the SQL commands will be read from standard input (your keyboard).

The retrieved data is in binary form and must be piped or redirected to some additional command filter/formatter to translate it into a more usable form (see the example below).

The fmltoasc command translates the binary output into ASCII format. The fmltoasc command formats the data in many different ways, depending on the options specified.

> **Important!** This command is to be considered deprecated in favor of Oracle database access methods, such as sqlplus, and will be removed in the future. Oracle is the NTM data store and data access via the Tuxedo database is to be discontinued.
>
> **Reference:** See the fmltoasc command (5-25) for more information on data formats.

Syntax

```
ongoing [-print option] [-f <strg>] [-m <strg>] [-s <char>]
   [-w <char>] [-n <char>] [-c <char>] [-i <strg>] [-t<n>] [-A] [-L]
   [-E] [file=file.sql]
```

> **Important!** The fmltoasc command options are now available in the ongoing command. Although fmltoasc is still available, we recommend that you use ongoing without the fmltoasc command. If no options (or only -A and/or -L) are used in the ongoing command, then you ***must*** use the fmltoasc command. However, if options are present, then the fmltoasc command ***cannot*** be used.

Before you begin

You may use the ongoing.test command (5-55) to check SQL files for errors before retrieving ongoing data.

Parameters

-print optionPrint options.

- p = Print the data in the form <fieldname>=<value>.
- r = Print only the retrieved fields, that is, those from the select clause of the SQL. The default is to print all the fields contained within the fielded buffer.

  Reference: "SQL command syntax" (p. 23)
- e = Print the value and exception level for calculated fields. The default is to print only the value.
- d = Use a predefined output format suitable for the DCS (Display Construction Set).
- h = Print output data in a column format with one header. The header will be printed, even if there isn't any data to return for the particular SQL.
- o = Print all the information held in the calc field in 5 columns:
  - print value — actual value of field
  - suspect — 0 = raw count not available; 8 = failure of raw count retrieval; 12 = couldn't perform indicated calculation properly; 14 = data suspect; 15 = data valid
  - incmplt — 0 = no, complete; 1 = yes, incomplete
  - x10 (times 10) — 0 = no, not times 10; 1 = yes, times 10
  - threshold level — actual threshold level for the field
- l = Print the label associated with the select clause. The label is optional in the SQL; therefore, this option may display a null string.

`-f <strg>` Use the string <strg> as the field delimiter string. The default delimiter string is the blank.

Important! Do not use the `-f` option with the `-w` and/or `-n` options because the -f option takes precedence.

`-m <strg>` Use the string <strg> as the message delimiter string. The default delimiter string is a newline.

`-s <char>` Use the character <char> as the subfield delimiter character. The default delimiter character is a blank. This option is used when a field is made up of several sub fields.

`-w <char>` Use the character <char> as the delimiter character before a string. The default delimiter character is a blank. This option is used mainly for formatting messages for the Display Construction Set User Interface Management System.

Important! Do not use the `-w` option with the `-f` option because the `-f` option takes precedence.

`-n <char>` Use the character <char> as the delimiter character before a number. The default delimiter character is a blank.

Important! Do not use the `-n` option with the `-f` option because the `-f` option takes precedence.

`-c <char>` Use the character <char> as the delimiter between fields in calculated fields. The default delimiter character is a blank.

`-i <strg>` Use the string <strg> as the invalid data string. The default invalid data string is “-1.”

`-t<n>` Use the number <n> as the time format definition, where:

- n = 1 *UNIX* date format
- n = 2 hour and minute
- n = 3 date, hour and minute
- n = 4 hour, minute, and second (default)
- n = 5 date, hour, minute, and second

`-A` Use to set the all fields option for the ENTDAT data type.

If you specify this option, the `ongoing` command will supply default values for fields that are not present. For example, if one of the fields you requested was `%occ` and `%occ` was not available for a certain record, then `ongoing` would fill in “-1” for its value.

If you do not specify this option, the `ongoing` command returns only fields that are in exception.

`-L` Use to set the late ignore option.

If you specify this option, the `ongoing` command will ignore all 5-minute data that occurs between the exc_upd event and the start of the next data collection period.

If you do not specify this option, the `ongoing` command will retrieve all “late” data.

`-E` Expand each discrete to include its DISCRETE_NAME, LABEL, and LEVEL, along with its current value.

This new -E option will return a -1 for the LABEL and LEVEL values in the following two cases:

1. The discrete is not valid for the office type of the selected record. The discrete will have a value of -2 in this case.
2. The select line did not request either OFFICE or TYPE. One of these two fields is necessary for `fmltoasc` to know the office type of the selected record so it can know which discrete file contains the LABEL and LEVEL.

`file.sql` External file

System responses

The system responds with error messages if any errors are encountered in interpreting the SQL commands. Otherwise, the output appears in binary form conforming to the *TUXEDO* FML.

> **Important!** If the `ongoing` command fails on the host, a pop-up window on the work station displays this message: “Process failed on host.” For more details about the problem, check the console window and the error log.

See the *TUXEDO FML Programmer's Guide* for detailed information on the binary output of this command. The binary output can be piped immediately to the `fmltoasc` command to produce ASCII output.

1. If no data is available in the shared memory, the `ongoing` command passes a “-1” on to the filter program.
2. Most of the data types are common to both the `ongoing` and `demand` commands, but the field names within those data types may vary for these two commands. If an SQL query for `ongoing` contains a data type in the FROM clause that is also valid for `demand` with field names that are only valid for `demand`, `ongoing` will not respond with error messages. Instead, it ignores those field names.

Examples

This is an example of an `ongoing` command line with options:

```
$ ongoing -r  tgctl.sql
```

In this command line, the contents of the SQL file “tgctl.sql” may appear as shown below.

```
SELECT  office sgtqs period
FROM    tgdat
WHERE   office = abcd or cmt = lyon1 or (etc)
```

This is an example of an `ongoing` command line piped to the `fmltoasc` command:

```
$ ongoing -AL  tgctl.sql | fmltoasc -r
```

References

“SQL command syntax” (p. 23)

Chapter 3, “SQL Interpreter” in the *Data Tables Guide*

Related command(s)

“demand” (p. 21)
“fmltoasc” (p. 25)
“urwformat” (p. 31)

□

ongoing.test

Description

The ongoing.test command can be used to check the system SQL files for errors. If an error is found, ongoing.test prints the appropriate error message. After you have corrected any errors, you may then use the ongoing command to retrieve data from the system's shared memory.

> **Important!** This command is to be considered deprecated in favor of Oracle database access methods, such as sqlplus, and will be removed in the future. Oracle is the NTM data store and data access via the Tuxedo database is to be discontinued.

Syntax

```
ongoing.test [file=file.sql]
```

Parameters

file.sql	You can place your system SQL commands in a file to be read by the ongoing.test command. If no file is specified, the SQL command interpreter reads commands from standard input. Press **CTRL - D** to terminate input.

System responses

If an error is found in the SQL file, ongoing.test quits and prints the appropriate error message. If no errors are found, the following message appears:

```
ongoing.test : No SQL errors found
```

Examples

```
$  ongoing.test  tgctl.sql
```

References

See Chapter 1, "All Data Fields" in the *Data Tables Guide* for more information on the retrievable fields.

Related command(s)

"demand" (p. 21)
"ongoing" (p. 51)

□

pas

Description

The pas (public announcement service) command displays the pas command (Public Announcement Service) data for a specific *4ESS* switch office for the current 5-minute period. The attempts and overflows for each pas command code are displayed.

Syntax

```
pas office=office
```

Parameters

office	Office name

System responses

Figure 16 shows an example of pas command output.

Figure 16 pas Command Output

```
IP
NM: 97/10/03 14:20      PAS REPORT okldca0349t
 ID  ANNOUNCEMENT          CODE          ATTEMPTS   OVERFLOWS

  0  stocks                9009761111         512         512
  1  weather               9009761212         511         511
  2  sports_results        9009761313         510         510
  3  time                  9009761616         509         509
  4  test1                 6148607765         508         508
  5  test2                 6148607765         507         507
  6  test2                 6148607765         506         506
  7  test4                 6148607765         505         505
  8  test5                 6148607765         504         504
  9  test6                 6148607765         503         503

REPORT COMPLETE
```

Examples

The output in Figure 16 was created by entering:

```
$ pas office=okldca0349t
```

References

See the “PAS Code File” (p. 64) in the *Record Base Administration Guide* for information on setting up the `pas` command reference data.

□

rbed

Description

The rbed (record base editor) command allows you to modify any record base file in the "/musr/rb" directory structure by placing you in a text editor of your choice.

> **Important!** Although you can select the text editor you wish to use, the editor of choice for NTM is the *Linux* System vi Text Editor.

Syntax

```
rbed [-r] file
```

Detailed description

The rbed command also "locks" other users out of the file while it is being edited, thereby preventing more than one user from working in a given file at the same time. The file remains locked until the first user exits rbed.

After using the rbed command, use the dbtest command to test the new changes, the create command to create the new changes in the database, and the installdb command to install the changes.

If an abnormal event such as a disconnection or a system crash occurs, you may need to recover an edited file. The *Linux* System vi Text Editor provides the "-r" option to perform this function. If the *Linux* System vi Text Editor is used, rbed supports the "-r" option to recover an edited file.

Parameters

-r	This option attempts to recover a file that was lost in an abnormal event (such as a disconnection or a system crash) while being edited.
file	File name of the record base file to edit (valid pathname is required if you are not in the directory where the file resides).

Examples

For example, to edit the Sets File:

- if you are in the "/musr/rb/sets" directory

```
$  rbed sets
```

- if you are not in the "/musr/rb/sets" directory

```
$  rbed /musr/rb/sets/sets
```

To recover a version of the edited Sets File

- if you are in the "/musr/rb/sets" directory

  ```
  $  rbed -r sets
  ```

- if you are not in the "/musr/rb/sets" directory

  ```
  $  rbed -r /musr/rb/sets/sets
  ```

System responses

Response when editing a file

There are three possible responses when the `rbed` command is used to edit a file. The following figures show the responses and explain the given options.

1. If the file is being edited by someone else (with the `rbed` command), it is considered "locked," and you receive a response similar to the following:

```
rbed: file /musr/rb/sets/sets is locked by user jsmith
   If lock is in error, have administrator delete
   /usr/spool/locks/rb00001a0a5008
```

   If the lock appears to be in error (i.e. an old lock resulting from a system problem, etc.), contact the System Administrator so that the lock can be removed manually.

2. If you choose to close (quit) the file and save the changes, you receive the following response:

```
Select action ([n]ew file, [s]ave & exit, save & [r]e-edit,
   [d]iscard & exit)
```

   These options are defined as follows:

   - n — create a copy of this file with a new filename. You will be asked to define the new filename.
   - s — save the changes to this file and exit the file
   - r — re-edit the file
   - d — discard the changes you made and exit the file

3. If you choose to close (quit) the file without saving the changes, you receive the following response:

```
File unchanged, select action ([n]ew file, [e]xit or [r]e-
   edit)
```

   These options are defined as follows:

   - n — create a copy of this file with a new filename. You will be asked to define the new filename.
   - e — exit the file without saving changes
   - r — re-edit the file

Response when recovering a file

After the `rbed` command is run with the "-r" option, the recovered file will appear on the screen.

Upon reviewing the recovered file, you will find that one of three possible degrees of recovery has taken place:

1. All of the changes you made were saved and recovered. If this is the case, save the changes and exit the file.
2. Some percentage of your changes were saved and some were lost. If this is the case, review the file and re-enter any changes that were not saved. Save the re-entered changes and exit the file.
3. None of the changes you made were saved. If this is the case, discard the recovered file. Re-edit the original file, save the changes and exit the file.

References

See Chapter 6, "Record Base Editor" in the *Record Base Administration Guide* for more information on the record base editor command and file locking.

See Chapter 5, "Record Base Files" in the *Record Base Administration Guide* for additional information on each record base file.

Related command(s)

"create" (p. 5)
"dbtest" (p. 17)
"installdb" (p. 34)

□

rdbstat

Description

The rdbstat (relational database status) command allows you to view the status of all the relational databases including those in the archive area.

Syntax

```
rdbstat
```

System responses

Figure 17 illustrates the output displayed on your screen after you run the rdbstat command.

Figure 17 rdbstat Command Output

```
$ rdbstat

     Historical Databases:

     --> start: 05/31/2005-00:00 end: 05/31/2005-23:55
     DB has NOT been backed up
     --> start: 06/01/2005-00:00 end: 06/01/2005-23:55
     DB has NOT been backed up
     --> start: 06/02/2005-00:00 end: 06/02/2005-23:55
     DB has NOT been backed up
     --> start: 06/03/2005-00:00 end: 06/03/2005-23:55
     DB has NOT been backed up
     --> start: 06/04/2005-00:00 end: 06/04/2005-23:55
     DB has NOT been backed up
     --> start: 06/05/2005-00:00 end: 06/05/2005-23:55
     DB has NOT been backed up
     --> start: 06/06/2005-00:00 end: 06/06/2005-23:55
     DB has NOT been backed up
     --> start: 06/07/2005-00:00 end: 06/07/2005-12:20

     Current Day

     Archive Databases:

     --> NM20050527_0005__20050527_0025: start: 05/27/2005-00:05 end: 05/27/2005
     --> NM20050528_0005__20050528_0105: start: 05/28/2005-00:05 end: 05/28/2005
     --> NM20050530_0000__20050530_0005: start: 05/30/2005-00:05 end: 05/30/2005
     --> NM20050605_1000__20050605_1000: start: 06/05/2005-10:00 end: 06/05/2000

* ARCHIVE DATABASES are databases that have been stored in the archive area.
```

References

See “The NTM data archive” (p. 9) in the *System Administration Guide* for more information about the NTM archive function.

recreate

Description

The `recreate` command places record base changes into the active database. It is executed automatically every night as part of the dayend procedure. It provides a simple way of incorporating record base changes into the active database. It determines which, if any, record base files have been modified since the last `installdb` (or `recreate`) command was run.

Syntax

```
recreate
```

Passwords / logins

If you are logged in with any login other than `nmadm`, you will be required to enter a password to execute this command.

If you are logged in as `nmadm`, no password is required.

If you have purchased Feature 22, “NMADM Login Accountability” you will not be prompted for the `nmadm` password.

When to use

This command will stop the system during execution, so it should only be executed at noncritical times.

No user should be editing a record base file when the `recreate` command is running.

This command will stop and later restart NTM if there are database files to incorporate into the active database.

It may also deactivate and reactivate offices while it performs a single office `create` ***on each one***.

Feature restrictions

If Feature 86, “Local Audit Data Restoration” is available, then the local audit is run for the office after the create process has finished. After all single-office creates are completed, the audit command is run for these offices to update the reference data.

If you have purchased Feature 8, “Disaster Recovery (Duplex)” and Feature 40, “Enhanced Disaster Recovery”, `recreate` will also synchronize the global record base files and record base partitions in the backup state from the primary host(s) before performing `create` or `installdb`. The audit results for offices in the backup mode will also be synchronized with those on the primary host(s).

Detailed description

For any record base files other than the ones associated with a single office create, the recreate command stops the system while it creates and installs the new files. After all modified files have been created and installed, the recreate command starts the system.

Executing the recreate command does not stop the system for the office, trunk group, and domain files. Instead, these files are updated with a single office create.

During a single office create, the office being created is deactivated. When the create process is finished, the recreate command activates the office and, if necessary, performs a reference audit to restore reference data lost during the create process.

If an office that is new to the database (has never collected data) was created by recreate, then the recreate command stops and starts the system a second time. If recreate performs a second stop/start of the NTM application, the following message appears:

```
Run <kill ${pid}> as root to disable this procedure.
```

Where pid is the process ID of the recreate command.

> **Important!** Usually, it is not necessary to kill the create process. One circumstance in which it may be desirable to do so is if dayend runs into a busy hour and you do not want the system to be stopped during that time.

If a number of offices must be recreated, the recreate process may take a long time. If you are performing an important task that requires the system to be running, you may kill the recreate process, using the appropriate process ID.

When the create process is finished, recreate activates the office. If necessary, the recreate command performs a reference audit to restore reference data lost during the create. The type of audit and whether or not it is executed is determined by the switch type created, as shown below. In addition, a local_audit is performed as needed.

Switch type(s)	Run by recreate
4ESS	tginfo
5ESS	pktsched+tgsched+tg+iec
1A *ESS*	ref
DMS, *EWSD*, or LSSGR	pktsched+tgsched+tg+iec
GTD-5	tgsched+tg

System responses

The recreate command identifies each command before it performs it. For example, if it is going to create a file, it will first print out: CREATING <filename> file.

In this example, <filename> is the record base file being created. The output from commands executed by the `recreate` command is also provided.

If `recreate` detects an error during a `create` or `installdb`, it will print out an error message and move on to the next file that needs to be created.

Figure 18 illustrates the output of the `recreate` command.

Figure 18 recreate command output

```
Password:
STOPPING system.
The system is now stopped.

CREATING rspte file
IP
Starting create at Fri Jun 15 11:30:08 1990
Creating rspte
Finished create at Fri Jun 15 11:31:51 1990

INSTALLING rspte file
IP
Install completed successfully

STARTING system.

                   NTM
   Copyright (c) 1990,1991,1992,1993,1994,1995,1996,1997 Lucent Technologies, Inc.

The system is now started.

DBTESTING office sim4e13ent01

DEACTIVATING office sim4e13ent01
  IP
sim4e13ent01 deactivated for audit discrete control meas

CREATING office sim4e13ent01
  IP
Have you run dbtest on office(s)? (yes, no)

Starting create at Fri Jun 15 11:34:40 1990
Initializing sim4e13ent01
Creating sim4e13ent01
        Adding 99 trunk groups to office
Finished create at Fri Jun 15 11:35:13 1990

ACTIVATING office sim4e13ent01
  IP
```

```
sim4e13ent01 activated for audit discrete control meas
RUNNING LOCAL_AUDIT RESTORE sim4e13ent01
Local Audit procedure starting for sim4e13ent01
See /musr/log/local_aud for the local audit output.
Start of local audit for sim4e13ent01 entity ...
Local audit for sim4e13ent01 entity completed.
Local Audit procedure completed
AUDITING sim4e13ent01 tginfo
  IP
sim4e13ent01 : BLK 47 : 4ESS TRUNK GROUP REFERENCE AUDIT
        sim4e14ent03-001 : added to active database schedule
        dba5e6ent13-002 : added to active database schedule
        sim4e14ent03-003 : added to active database schedule
        sim4e14ent03-004 : added to active database schedule
        dba5e6ent13-005 : added to active database schedule
        dba5e6ent13-006 : added to active database schedule
        sim4e14ent03-007 : added to active database schedule
        sim4e14ent03-008 : added to active database schedule
        sim4e14ent03-009 : added to active database schedule
        sim4e14ent03-010 : added to active database schedule
        sim4e14ent03-011 : added to active database schedule
        sim4e14ent03-012 : added to active database schedule
        sim4e14ent03-013 : added to active database schedule
        sim4e14ent03-014 : added to active database schedule
        sim4e14ent03-015 : added to active database schedule
sim4e13ent01 : BLK 47 : 4ESS MANUAL TRUNK GROUP CONTROL AUDIT
sim4e13ent01 : BLK 47 : 4ESS AUTOMATIC TRUNK GROUP CONTROL AUDIT

DATABASE is up-to-date with latest record base changes.
$
```

References

Chapter 7, “Record Base Administration” in the *Record Base Administration Guide*

Related command(s)

“act” (p. 4) “installdb” (p. 34)
“audit” (p. 16) “local_audit” (p. 22)
“create” (p. 5) “startsys” (p. 23)
“deact” (p. 11) “stopsys” (p. 28)

□

tg_grep

Description

The tg_grep (duplicate trunk group set check) command searches the database for trunk group files containing a given trunk group set name and displays them on your screen. When you want to define a new trunk group set, use this command to make sure the set name is not already in use.

Syntax

```
tg_grep [set=tgset]
```

Feature restrictions

With BDR installed, the tg_grep command will search through all the record base partitions when necessary.

Restrictions

This command cannot be used by more than one person at a time. If you attempt to execute this command when it is already in use, you will receive a message asking you to try again later.

Parameters

tgset	Trunk group set name

System responses

Figure 19 illustrates the output for the tg_grep command.

Figure 19 tg_grep Command Screen Output

```
Enter the desired TG SET: cb2

The set cb2 already exists in the files:

/musr/rb/tg/clmboh0001t
/musr/rb/tg/clmboh0002t
/musr/rb/tg/clmboh0003t
/musr/rb/tg/clmboh0004t
/musr/rb/tg/clmboh0005t
/musr/rb/tg/clevoh0003t
/musr/rb/tg/clevoh0004t
/musr/rb/tg/clevoh0007t
/musr/rb/tg/clevoh0008t
/musr/rb/tg/clevoh0009t
```

```
/musr/rb/tg/clmboh0049t
/musr/rb/tg/clmboh0050t
/musr/rb/tg/clmboh0006t
/musr/rb/tg/clmboh0009t
/musr/rb/tg/clmboh0010t

The set cb2 already exists in the /musr/rb/sets/sets file.
```

Examples

To generate the output in Figure 19 without waiting for the system to prompt you for a trunk group set name, you can enter the command as follows:

```
$  tg_grep  cb2
```

Related command(s)

"ofc_grep" (p. 47)

□

tgset

Description

The `tgset` (trunk group set check) command retrieves and displays all the trunk groups in a given set. To ensure that the set name you enter is valid, check it first in the "/musr/rb/sets/sets" file.

Syntax

```
tgset setname=setname from_office=from_office to_office=to_office
```

Feature restrictions

With BDR installed, the `tgset` command will search through all the record base partitions when necessary.

Restrictions

This command cannot be used by more than one person at a time. If you attempt to execute this command when it is already in use, you will receive a message asking you to try again later.

Parameters

`setname`	Name of the desired trunk group set
`from_office`	Name of desired From-office or "all"
`to_office`	Name of desired To-office or "all"

System responses

Figure 20 illustrates the output for the `tgset` command.

Figure 20 tgset Command Screen Output

```
$ tgset
setname= cb2
from_office= anhmca11ds0
to_office= all


From_Office      To_Office     Suffix
-----------      ---------     ---------
anhmca11ds0      almdca1176a   a001
anhmca11ds0      almdca1176a   a002
```

```
End tgset at:  Wed Jun 21 14:40:07 EDT 1995
```

Examples

To generate the output above without waiting for the system to prompt you for a trunk group set name, you can enter the command as follows:

```
$ tgset cb2 anhmca11ds0 all
```

Related command(s)

"ofcset" (p. 49)

□

thrconvert

Description

Use the `thrconvert` (threshold convert) command to convert old (static) threshold or filter files into the new rules-based format used by the database for Feature 189, "Replacement Thresholding Capability for Trunk Group Data". This command is available only with this feature.

Syntax

```
thrconvert filetype inputfile outputfile
```

Important! The `thrconvert` command looks for commas as delimiters between keyword phrases in the threshold and filter files. If you do not have commas as delimiters, this command may fail.

Cautions

Please read the following cautions carefully.

Following the conversion of a Trunk Group Threshold File or the Filter File to the new rule-based thresholding format available with the purchase of Feature 189, "Replacement Thresholding Capability for Trunk Group Data", modifications can be made to the thresholding rules created in the "outputfile" with the new rule syntax.

However, for the converted threshold rules to be integrated into the system via the `create` or `thresh` commands, the user must overwrite the existing thresh[n] or filter file with the "outputfile" designated in the `thrconvert` command.

If the user desires to retain the old syntax trunk group threshold file or filter file for future reference, the user must first change the file name of the old file, or move the old file to another directory, before overwriting the old file with the "outputfile" specified in the `thrconvert` command.

Parameters

`filetype`	thresh, ATM or filter, indicating the trunk group threshold, ATM threshold, or the filter files.
`inputfile`	Name of the file to be converted
`outputfile`	Name of the rule-based file generated by this command

System responses

Figure 21 shows the response for the `thrconvert` command.

Figure 21 thrconvert system response

IF the command is …	THEN the system responds with …
executed successfully	...... start conversion parse .... end conversion parse .....
entered incorrectly	Usage: thrconvert thresh/filter input output
unsuccessful	The position of the error and “Bad input file, the output file is not correct.”

Examples

```
$ thrconvert thresh /musr/rb/thresh/thresh1 thresh1_rule
$ thrconvert filter /musr/rb/thresh/filter filter_rule
```

References

“RSPTE File” (p. 68) in the *Record Base Administration Guide*

□

thresh

Description

Use the thresh (threshold) command to create a threshold table from a record base threshold file. The thresh command modifies the threshold table for the threshold file specified. Only the indices that exist in the file are modified. This command updates the current database.

If a particular index is duplicated in the referenced thresh files, the index from "/musr/rb/thresh/thresh#" will be installed and an error message will be generated for the duplicate.

Syntax

```
thresh file=file
```

Passwords / logins

To run this command user must be a member of the 'rb' group.

Before you begin

Before attempting to install the new thresh file, the "/musr/rb/thresh/sched" file must be updated to include the new thresh file.

Feature restrictions

BDR

For systems with Feature 8, "Disaster Recovery (Duplex)" and Feature 40, "Enhanced Disaster Recovery" the thresh command will modify the threshold table in the current database from the file "/musr/rb/thresh/thresh#" and the threshold file in the other valid record base partitions "/musr/snw/*partition_name*/rb/thresh/thresh#", where # is a value of 1–8.

> **Important!** If BDR is installed and active, the record base files will be transferred to and modified on all the appropriate BDR hosts.

If a particular index is duplicated in the referenced thresh files, the index from "/musr/rb/thresh/thresh#" will be installed and an error message will be generated for the duplicate.

Partitions and Subnetworks

For systems with Feature 3, “Management of Record Base Partitions and Subnetworks”, the `thresh` command processes thresh[n] files in the main partition first, then the subnetwork/partition thresh[n] files. If duplicate indexes are found, the indexes defined in the main partition have precedence over those in the subnetworks.

Parameters

`file`	Valid threshold file name or enter (atm, packet)

Notes: For UDDM use file name in format: <uddm>_thresh<1-8>

System responses

Figure 22 shows the response for the `thresh` command.

Figure 22 thresh system response

IF the command …	THEN the system responds with …
executes successfully	IP Creating thresh1
generates errors	*`error item specific`*

Where:

`error`	duplicate, invalid
`item`	calculation, flag, or index
`specific`	the ID of the *`item`*

Notes:

- Duplicate calculations are processed only once. Subsequent instances of the same calculation are ignored.
- An index is processed only the first time it is found within the thresh[n] file.
  - Duplicate indexes within the same file are skipped.
  - Duplicate indexes across partitions or subnetworks for the same thresh table are skipped.
- Invalid calculations or flags are ignored. `create` prints an error message for each invalid item and continues processing the file, ignoring the line with the error.

Examples

```
$ thresh file=thresh1
```

References

See the “Trunk Group Threshold File” (p. 98) and Chapter 8, “Performing a full create and installdb” in the *Record Base Administration Guide*.

Related command(s)

“create” (p. 5)

□

upd_meta

Description

Use the `upd_meta` command to change metadata for:

1. Labels for legacy types on Periodic Data Browser page. The user-defined labels list is in the *"/nm/dbutil/metadata/labels.U"* file.
2. Aggregation formula for legacy types. The aggregation expression list is in the *"/nm/dbutil/metadata/aggmode.U"* file.

Syntax

```
upd_meta
```

Before you begin

Before running `upd_meta` command edit the *"/nm/dbutil/metadata/labels.U"* or the *"/nm/dbutil/metadata/aggmode.U"* file. Remember to always extend files *"/nm/dbutil/metadata/labels.U"* and *"/nm/dbutil/metadata/aggmode.U"* with new data to prevent from losing previously written data.

Labels File Example

```
%,TABLENAME,FIELDNAME,LABEL
LINKDAT,ENT_ID,’Switch’
LINKDAT,SLS_NAME,NULL
```

Aggmode File Example

```
Aggmode File Example
%,TABLENAME,FIELDNAME,AGGMODE
ENTDAT_GSP,TOT_CT,'sum(ENTDAT_GSP.TOT_CT)'
TGDAT,TG_BIDS,'max(TGDAT.TG_BIDS)'
TGDAT,WBAND,NULL
```

Parameters

`TABLENAME` Table name.

`FIELDNAME` Field name.

`LABEL` User-defined label provided in ‘’ signs or NULL value for removing custom label and restore the default.

`AGGMODE` Any valid oracle aggregation expression .

System responses

Figure 23 shows the response for the `upd_meta` command.

Figure 23 upd_meta system response

IF the command ...	THEN the system responds with ...
Labels executes successfully	Some values of 'label' attribute of userfieldschema table can be lost. Continue ? [yn]y Updating... Successfully updated 1 rows in userfieldschema table.
Aggmode executes successfully	Some values of 'aggmode' attribute of fieldschema table can be lost. Continue ? [yn]y Testing... ENTDAT_GSP,TOT_CT,'sum(ENTDAT_GSP.TOT_CT)' Updating... Successfully updated 1 rows in fieldschema table.
generates error because the Fieldname or Tablename does not exists	merge into nm$.userfieldschema u * ERROR at line 1: ORA-02291: integrity constraint (NM$.USERFIELDSCHEMA_FIELDNAME) violated - parent key not found Could not update userfieldschema.

Examples

```
$ upd_meta
```

References

See the “Periodic Data Browser container page” (p. 8-27) in the *User Guide*.

6 Report Commands

Overview

Purpose

This chapter lists and describes the NTM report commands. Use report commands to modify, generate, or output reports.

> **Important!** These commands are ***not*** related to Feature 272, "NTM Report Writer". For information on that feature, see the *Report Writer Guide*.

Contents

This chapter discusses the following topics:

□

Command functions

Table

Table 1 lists each NTM report command and its function.

Table 1 Report commands and functions

Command	Function
`darpt`	Creates Detailed Analysis Report
`failrep`	Outputs reports of switches with late measurement data or unsuccessful discrete collection
`firpt`	Creates Final Trunk Group Report
`icrpt`	Produces the Idle Capacity Report
`ncrpt`	Creates Daily No Circuit Report
`oprpt`	Creates Office Performance Report
`perfrep`	Outputs performance reports based on historical data
`urwformat`	Produces user report writer format

□

darpt

Description

The darpt (detailed analysis report command) provides a listing of all high-usage trunk groups that could have an impact on a specific final trunk group for a given hour. The Detailed Analysis Report lists all high-usage trunk groups that have a minimum *pofl* greater than or equal to the value entered and a minimum *oflpc* greater than or equal to the value entered. The report provides data only on the high-usage trunk groups that could impact final trunk groups between the offices provided.

Syntax

```
darpt from=A_office to=B_office pofl=pofl oflpc=oflpc [hour=hour]
```

Parameters

A_office	”From Office” Name
B_office	”To Office” Name
pofl	Minimum pofl (percent overflow) of HU (high usage) trunk groups. HU trunk groups having a pofl greater than or equal to this value are displayed.
oflpc	Minimum oflpc (overflow peg count) of HU trunk groups. HU trunk groups having an oflpc greater than or equal to this value are displayed.
hour	Start time of the hour you are requesting. The format for entering the hour is YY/MM/DD-HH:mm.

Important! If you do not specify an hour, the preceding current network hour is used. The hour you specify must be on a 1-hour boundary, for example, 13:00.

System responses

Figure 1 shows the responses for the darpt command.

Figure 1 darpt command responses

IF the command is ...	THEN the system responds with ...
executed successfully	REPORT WRITER -- reporting by file '/nm/reports/darpt_exe' Sending output to file '/usr/tmp/*tempfile*' Processing query. **...** Retrieving data from Database. **...** Executing report. **...** Report '/nm/reports/darpt_exe' completed The darpt report can be found in /musr/admin/output_darpt

When the `darpt` command is completed, the output is appended to the *"output_darpt"* file in the "/musr/admin" directory. Use the *Linux* system `pg` command to view the contents of the file. Remember to clean out this file periodically.

Figure 2 shows an example of a report generated by the `darpt` command.

> **Important!** This report example is not intended to be an exact duplicate of the report provided by NTM. It is an example report that can be customized to the users' needs.

Figure 2 Detailed Analysis Report (darpt)

```
DETAILED ANALYSIS REPORT: NTM
DATE:                     24-JAN-1994

THRESHOLDS:          NWT:  = 0800
                 %OFL HU: >= 10.0
               HU OFL PC: >= 200.0
                                                              PAGE    1

FROM           TO             SUF  1WIN 2WAY  1WO ALT              PER
NWT    PC  OFL %OFL  CCS/TRK %OCC  IPC ICCH ACH OCCH RRATT RRSUC    COL

OKLDCA0349T    BRLNCA0134G   P7        0  200    0 SNFCCA2143T   P7   12
0800 5892 5904  100      2.7 **** 5916    0   0    0
OKLDCA0349T    COLACA0199H   P7        0  100    0 SNFCCA2143T   P7   12
0800 2400  624   26      6.5   56  624    6  25   19
OKLDCA0349T    IGNCCA1288L   2P        0  200    0 SNFCCA2143T   P7   12
0800 3252 3264  100      1.5 **** 3276    0   0    0
OKLDCA0349T    LRKSCA11DS0   2P        0  100    0 SNFCCA2143T   2A   12
0800  624  624  100      6.5   56    0    0   6    0
OKLDCA0349T    LSATCA11DS0   P7        0  100    0 SNFCCA2143T   P7   12
0800 2400  624   26      6.5   56  624    6  25   19
OKLDCA0349T    MLBRCA11DS0   2P        0  100    0 SNFCCA2143T   P7   12
```

```
0800 2400  624   26      6.5   56  624    6  25   19
OKLDCA0349T    MLVYCA0138R   2P       0  200    0 SNFCCA2143T   P7   12
0800 5652 5664  100      2.6 **** 5676    0   0    0
OKLDCA0349T    MTVWCA1196G   P7       0  100    0 SNFCCA2143T   P7   12
0800 2400  624   26      6.5   56  624    6  25   19
OKLDCA0349T    MTVWCA11DS0   PH       0  200    0 SNFCCA2143T   P7   12
0800  252  264  104      0.1 1651  276    1   1    0
OKLDCA0349T    PCFCCA11DS0   PH       0  100    0 SNFCCA2143T   P7   11 *
0800 2400  624   26      6.5   56  624    6  25   19
OKLDCA0349T    PLALCA1249C   P7       0  100    0 SNFCCA2143T   P7   12
0800 2400 2400  100      6.5   56  240    2  25    0
OKLDCA0349T    RDCYCA0136A   2P       0  100    0 SNFCCA2143T   2A   12
0800 8412 8424  100      3.8 **** 8436    0   0    0
OKLDCA0349T    SNBUCA0258K   P7       0  100    0 SNFCCA2143T   P7   12
0800 2400  624   26      6.5   56    0    0  26   19
OKLDCA0349T    SNBUCA02DS0   P7       0  100    0 SNFCCA2143T   P7   12
0800 2400  624   26      6.5   56    0    0  25   19
OKLDCA0349T    SNCRCA1159A   P7       0  100    0 SNFCCA2143T   P7   10 *
0800 2400  624   26      6.5   56    0    0  25   19
OKLDCA0349T    SNFCCA0133M   P7       0  100    0 SNFCCA2143T   P7   12
0800 2400  624   26      6.5   56    0    0  25   19
OKLDCA0349T    SNFCCA0136A   P7       0  100    0 SNFCCA2143T   P7   12
0800 2400  624   26      6.5   56    0    0  25   19
OKLDCA0349T    SNFCCA0195M   P7       0  100    0 SNFCCA2143T   P7   12
0800 2400  624   26      6.5   56  624    6  25   19
OKLDCA0349T    SNFCCA0198A   P7       0  100    0 SNFCCA2143T   P7   12
0800 2400  984   41      6.5   56  996   10  25   15
OKLDCA0349T    SNFCCA01DS0   2P       0  200    0 SNFCCA2143T   2A   12
0800 4932 4944  100      2.2 **** 4956    0   0    0

        *  These entries were calculated on less than 12 periods
           of data.  The number represents the actual number of
           periods that were available.
```

Table

Table 2 explains the headers in Figure 2.

Table 2 Description of Headers in Figure 2

LABEL	MEANING
FROM	From Office Name
TO	To Office Name
SUF	Trunk group suffix
1WIN	Number of 1-way incoming trunk groups
2WAY	Number of 2-way trunk groups

Table 2 Description of Headers in Figure 2

LABEL	MEANING
1WO	Number of 1-way outgoing trunk groups
ALT	CLLI and suffix of alternate trunk group
NWT	Current hour and minute
PC	Peg count
OFL	Overflow
%OFL	Percentage of overflow
CCS/TRK	Number of CCS (Common Channel Signaling) trunks
%OCC	Percentage of occupancy
IPC	Incoming peg count
ICCH	Incoming connections per circuit per hour
ACH	Attempts per circuit per hour
OCCH	Outgoing connections per circuit per hour
RRATT	Number of reroute attempts
RRSUC	Number of successful reroute attempts
PER COL	Number of periods

Examples

The following is an example of a `darpt` command entry:

```
$  darpt  from=shokca0589t to=chcgilcl57t pofl=35 oflpc=50
   hour=13:00
```

If executed after 14:00, this entry produces a Detailed Analysis Report for office “shokca0589” for the twelve 5-minute periods starting at 13:00 of the day the command was executed. The report contains only those entries that:

- Exceed or equal 35 percent overflow
- Have an overflow peg count greater than or equal to 50

Figure 3 is an example of an entry you can add to a crontab file to produce a Detailed Analysis Report each Monday through Friday for the hour beginning at 13:00.

Figure 3 Entry added to crontab file to produce Detailed Analysis Report

```
5 14 * * 1-5 . /etc/nm.env; /nm/cmdbin/darpt shok chcg 0 0
```

You can change the example shown in Figure 3 to produce a user-modified version of the report (assuming there is a modified darpt_exe report data file in the directory <directory name>). That entry is shown in Figure 4.

Figure 4 Entry added to crontab file to produce Modified Detailed Analysis Report

```
5 14 * * 1-5 . /etc/nm.env; cd <directory name>;
   /nm/cmdbin/darpt shok chcg 0 0
```

References

Chapter 4, "User Report Writer" in the *Data Tables Guide* for more information on modifying example reports.

□

failrep

Description

The `failrep` command is used to output two types of collection reports:

- A report on switches that do not report their 5-minute measurement data or are late in reporting
- A report on switches for which collection of discrete data was not successful during the 30-second discrete collection cycle.

Report output begins with the current cycle and continues until either the number of periods specified in the command entry is reached, or the command is cancelled with the **DEL** key.

Syntax

```
failrep [argument] [num_of_periods]
```

Important! Notice that name-defined keywords are not used in this command.

Parameters

`argument` Valid values are:

- dsc — Discrete report, which lists each switch for which discrete data collection was not successful during 30-second collection cycles. A resp_code (response code) message indicates the reason for individual failures.
- ofc — Office report, which lists the names of offices (switches) that report their 5-minute measurement data late or do not report their data at all. For each late-reporting switch, the report lists the status (number of seconds the data was late), and the date and time of the collection cycle.

Since the `failrep` command with the `ofc` argument takes 5 minutes to display results for a measurement collection cycle, some system events are displayed to let the user know the command is active. These events are:

- Start of Exception Processing — The beginning time of the collection cycle.
- Discrete End of Period — Repeated with each 30-second occurrence.
- Exception End of Period — The time in the collection cycle when displays in autoupdate mode are updated with exception data from shared memory. Measurement data received and processed after this time is considered late.

`num_of_periods` The number of periods (collection cycles) for which the `dsc` or `ofc` reports should be printed to the screen. The default is 12. For a `dsc` output, this is twelve 30-second cycles, or 6 minutes. For an `ofc` output this is twelve 5-minute collection cycles, or 1 hour.

System responses

Figure 5 illustrates the `failrep` command report output when the `dsc` argument is specified. The command entered to produce this report was:

```
$ failrep dsc 3
```

The failrep `dsc` report receives its input by using the ongoing process to read shared memory. In the first iteration of the example report, shared memory contained the names and response code messages for six switches for which discrete collection was not successful. These names are not repeated in subsequent output, although the switches may be still OOS (out of service). Only new collection failures will be listed. If no switches have failures, then only the report header indicating the Discrete End of Period time is printed.

Figure 5 Data Collection Failure Report

```
NM Data Collection Failure Report

Collection Started:  Mon Aug 27 09:18:23 1994
             Ended:  Mon Aug 27 15:34:45 1994


                        Five-minute                        30-second
Office                Failures        Timeouts          Failures       Timeouts
======                ========        ========          ========       ========

sim4e13ent01             1               3                 0              1
sim4e12ent02             0               0                 1              0
sim4e12ent03             0               1                 1              0
```

Table 3 explains the headers used in Figure 5.

Table 3 Description of headers in failrep dsc report

Label	Meaning
office	The CLLI name of the switch for which discrete collection was not successful
rsp_code	The responsibility code message indicating the reason for the failure to collect discretes. For the complete list of possible responsibility code messages see Table 2, "Link expansion display status messages" (p. 5) in the *Data Tables Guide*.
date	The year, month, and day of the collection cycle

Table 3 **Description of headers in failrep dsc report**

Label	Meaning
time	The end of period time of the 30-second discrete collection cycle

Examples

Use the following command line to generate a report similar to the one shown in Figure 5.

```
$ failrep dsc 3
```

References

See the `perfrep` command (6-27) for more information on collecting data collection failure statistics.

For more information on the `failrep` and `perfrep` commands and reports, see Chapter 7, "Administrative Performance Reports" in the *System Administration Guide*.

Related command(s)

"perfrep" (p. 27)

□

firpt

Description

The `firpt` command produces the Final Trunk Group Report, providing a list of all final trunk groups for a specific “From Office” and hour. This report shows peg counts, reroute, and overflow information.

Syntax

```
firpt from=office [hour=hour]
```

Parameters

`office` ”From Office” name

`hour` Start time of the hour requested. The format for entering the hour is YY/MM/DD-HH:mm.

Important! Specify the hour on an even 1-hour boundary, for example, 13:00. If you do not specify an hour, the preceding current network hour is used.

System responses

Figure 6 shows the responses for the `firpt` command.

Figure 6 firpt command responses

IF the command is ...	THEN the system responds with ...
executed successfully	REPORT WRITER -- reporting by file /nm/reports/firpt_exe Sending output to file /usr/tmp/*tempfile* Processing query. **...** Retrieving data from Database. **...** Executing report. **...** Report '/nm/reports/firpt_exe' completed The firpt report can be found in /musr/admin/output_firpt

When the `firpt` command is completed, the output is appended to the *“output_firpt”* file in the “/musr/admin” directory. Use the *Linux* system `pg` command to view the contents of the file. Remember to clean out this file periodically.

Use the crontab entry shown in Figure 7 to:

- Send the contents of the output file to the printer
- Clear the file
- Remove the entry at specified intervals

Figure 7 Crontab entry used with firpt command

```
0 0 1 * * lp -c /musr/admin/output_firpt;
  >/musr/admin/output_firpt
```

Figure 8 shows an example of a report generated by the `firpt` command.

> **Important!** This report example is not intended to be an exact duplicate of the report provided by NTM. It is an example report that can be customized to the users' needs.

Figure 8 Final Trunk Group Report

```
FINAL TRUNK GROUP REPORT: NTM
        DATE:                     27-AUG-1994

        THRESHOLD:       MACHINE: GRDNCA0211T
                        NWT HOUR: 1400

               PERIODS COLLECTED: 12
                                                                 PAGE     1


                                                                       PER
        TO OFFICE     SFX     PC    OFL   %OFL   %OCC    RRATT   RRSUC COL

        ANHMCA0211T    FC   1412      0      0     20        0       0  11 *
        ARTSCAXF40KI   FA   1279      0      0     70        0       0  12
        ARTSCAXF86KI   FA   1726      0      0     45        0       0  12
        AVINCA11CGO    FA    241      0      0     39        0       0  12
        AVINCA11CGOL   FA    383      0      0     27        0       0  12
        BEIICA1177E    CF    409      0      0     27        0       0  12
        BEIICA1177EI   FA   1126      0      0     42        0       0  12
        BEIICA11CGO    CF    207      0      0     22        0       0  12
        BEIICA11CGOI   FA    155      0      0     57        0       0  12
        BIFICAXF86HI   FA    336      7      2     48        0       0  12
        BIGRCAXF92AI   FA    182      0      0     22        0       0  12
        CICYCA1183A    CF    576     34      5     88        0       0  12
        CMTNCA01CG1    CF   1606     82      5     89        0       0  12

        *  These entries were calculated on less than 12 periods
           of data.  The number represents the actual number of
           periods that were available.
```

Table

Table 4 explains the labels in Figure 8.

Table 4 Description of headers in Final Trunk Group Report

LABEL	MEANING
TO OFFICE	To Office name
SFX	Trunk group suffix of up to 4 digits
PC	Peg count
OFL	Number of overflows
%OFL	Percentage of overflow
%OCC	Percentage of occupancy
RRATT	Number of reroute attempts
RRSUC	Number of successful reroutes
PER COL	Number of periods

If you add the entry shown in Figure 9 to crontab, a Final Trunk Group Report is produced each Monday through Friday for the hour beginning at 13:00.

Figure 9 Entry used with crontab to produce Final Trunk Group Report

```
5 14 * * 1-5 . /etc/nm.env; cd <directory name>;
   /nm/cmdbin/firpt nrwy
```

Examples

The following is an example of an `firpt` command entry:

```
$  firpt  from=nrwyilno41t hour=14:00
```

References

See Chapter 4, “User Report Writer” in the *Data Tables Guide* for more information on modifying example reports.

□

icrpt

Description

The icrpt command produces the Idle Capacity Report, which lists all trunk groups that have exceeded given thresholds for a specific office and hour. The Idle Capacity Report lists all trunk groups that have *pocc* less than or equal to the entered value and trunk groups having circuit counts greater than or equal to the entered values.

Syntax

```
icrpt from=office pocc=pocc trks=trks [hour=hour]
```

Parameters

office ”From Office” Name.

pocc Percent occupancy threshold. Trunk groups having a *pocc* less than or equal to this value are displayed on the report.

trks Number of trunks (1- and 2-way) in the trunk group. Trunk groups having circuit counts greater than or equal to this value are displayed on the report.

hour Start time of the hour you are requesting. The format for entering the hour is YY/MM/DD-HH:mm.

Important! Specify the hour on an even 1-hour boundary, for example, 13:00. If you do not specify an hour, the preceding current network hour is used.

System responses

Figure 10 shows the responses for the icrpt command.

Figure 10 icrpt command responses

IF the command is ...	THEN the system responds with ...
executed successfully	REPORT WRITER -- reporting by *file* /nm/reports/icrpt_exe Sending output to file /usr/tmp/*tempfile* Processing query. **...** Retrieving data from Database. **...** Executing report. **...** Report /nm/reports/icrpt_exe completed The icrpt report can be found in /musr/admin/output_icrpt

When the `icrpt` command is completed, the output is appended to the *"output_icrpt"* file in the "/musr/admin" directory. Use the *Linux* system `pg` command to view the file. Remember to clean out this file periodically.

Use the crontab entry in Figure 11 to produce an Idle Capacity Report each Monday through Friday for the twelve 5-minute periods beginning at 13:00:

Figure 11 Crontab entry used with the icrpt command

```
5 15 * * 1-5 . /etc/rnms.env; /nm/cmdbin/icrpt shok 95 12
```

You can change the entry shown in Figure 11 to produce a modified version of the report (assuming there is a modified icrpt_exe report data file in the directory <*directory name*>. The modified entry is shown in Figure 12.

Figure 12 Modified crontab entry used with icrpt

```
5 15 * * 1-5 . /etc/rnms.env; cd <directory name>;
   /nm/cmdbin/icrpt office=shok 95 12
```

Figure 13 shows an example of a report generated by the `icrpt` command.

> **Important!** This report example is not intended to be an exact duplicate of the report provided by NTM. It is an example report that can be customized to the users' needs.

Figure 13 Idle capacity report

```
IDLE CAPACITY REPORT: NTM
DATE:                 27-AUG-1994

THRESHOLDS:  MACHINE: SHOKCA0263t
            NWT HOUR: 1400
                %OCC: <= 95
```

```
     TRUNK QUANTITY: >= 12
                                                 PAGE    1

                                                        PER
TO OFFICE     SUF            %OCC          TRUNK QUANTITY COL

LNCSCAXG52T   FA                4                66       11 *
CXNRCAXF53T   FA                1                90       12
SNBBCAXF54T   FA                9               154       12
SNTMCAXF53T   FA               52                96       12
VNTRCA0258T   HC                0                52       12
CKIDCA0344T   HC               57               228       12
MDSTCA0243T   HB               89                41       12
PROVUTMA01T   HC               92                12       12
CRCHTXTU04T   HC               95                12       12
FIPSTXMA02T   HC               90                12       12
CNCNOHWS14T   HC               64                24       12
CLMBOH1103T   HC               95                12       12
DYTNOH1501T   HC               91                12       12
WCHTKSBR24T   HC               95                12       12
ATLNGANW05T   HC               91                12       12
HRFRCT0304T   HC               85                12       12
RCPKNJ0203T   HC               74                24       12
NYCMNY5450T   HC               93                36       12

*  These entries were calculated on less than 12 periods
   of data.  The number represents the actual number of
   periods that were available
```

The report contains entries that have the following:

- A percentage of occupancy of less than or equal to 95
- A trunk count greater than or equal to 12

Table

Table 5 explains the labels in Figure 13.

Table 5 Description of headers in Idle Capacity Report

LABEL	MEANING
TO OFFICE	To Office Name
SUF	Trunk group suffix of up to 4 digits
%OCC	Percentage of occupancy
TRUNK QUANTITY	Number of trunk groups
PER COL	Number of periods

Important! For this example, do not execute this command before 15:00. The data required to produce the report is not present in the database before this time.

Examples

The following is an example of an `icrpt` command entry:

```
$  icrpt  from=shokca0263t pocc=95 trks=12 hour=16:00
```

References

See Chapter 4, “User Report Writer” in the *Data Tables Guide* for more information on modifying example reports.

□

ncrpt

Description

The `ncrpt` command produces the Daily No Circuit Report, which lists all the final trunk groups that have exceeded given thresholds for a specific office and hour.

Syntax

```
ncrpt from=office pofl=pofl ccs/trk=ccs/trk pocc=pocc [hour=hour]
```

Parameters

`office` ”From Office” Name

`pofl` Minimum percent overflow of the final trunk group. Final trunk groups having an `pofl` greater than or equal to this value are displayed on the report.

`ccs/trk` Per trunk group usage measurement based on CCS (100 Call Seconds). Final trunk groups having *ccs/trk* greater than or equal to this value are displayed on the report (1 CCS = 1 circuit in use for 100 seconds).

`pocc` Minimum percent occupancy allowed. Final trunk groups having a `pocc` greater than or equal to this value are displayed on the report.

`hour` Start time of the hour being requested. The format for entering the hour is YY/MM/DD-HH:mm.

Important! Specify the hour on an even 1-hour boundary, for example, 13:00. If you do not specify an hour, the preceding current network hour is used.

System responses

Figure 14 shows the responses for the `ncrpt` command.

Figure 14 ncrpt command responses

IF the command is ...	THEN the system responds with ...
executed successfully	REPORT WRITER -- reporting by file '/nm/reports/ncrpt_exe' Sending output to file '/usr/tmp/*tempfile*' Processing query. **...** Retrieving data from Database. **...** Executing report. **...** Report '/nm/reports/ncrpt_exe' completed The report can be found in /usr/admin/output_ncrpt

When the `ncrpt` command is completed, the output is appended to the *"output_ncrpt"* file in the "/musr/admin" directory. Use the *Linux* system `pg` command to view the contents of the file. Remember to clean out this file periodically. You can add the entry shown in Figure 15 to crontab to produce a Daily NC (No Circuit) Report Monday through Friday for twelve 5-minute periods beginning at 14:00:

Figure 15 Crontab entry used with ncrpt command to produce Daily NC Report

```
5 14 * * 1-5 . /etc/rnms.env; /nm/cmdbin/ncrpt all 0.5 27.0
   75.0
```

You can change this entry to produce a user-modified version of the report (assuming there is a modified `ncrpt_exe` report data file in the directory <*directory name*>). The new entry is shown in Figure 16.

Figure 16 Crontab entry used with ncrpt command to produce Modified Daily NC Report

```
5 14 * * 1-5 . /etc/rnms.env; cd <directory name>;
   /nm/cmdbin/ncrpt all 0.5 27.0 75.0
```

Figure 17 shows an example of report generated by the `ncrpt` command.

Figure 17 Daily No Circuit Report

```
DAILY NC REPORT: NTMS
  DATE:            01-JUN-1994

  THRESHOLDS:
             NWT:   =  1400
            %OFL: >=  0.5
         CCS/TRK: >= 27.0
            %OCC: >=  7.5
                                                                PAGE    1

  FROM          TO            SUF    1WIN 2WAY  1WO ALT             PER 5-MIN
```

```
NWT    PC  OFL %OFL CCS/TRK %OCC  IPC ICCH ACH OCCH RRATT RRSUC COL OFL

ANHMCA0211T   BRFACA1252T    FA      0    0    0                    ---+--
0800  645    2  0.5    35.9   84    0    0   4    4     0     0  12 ------

ISANCA0470T   ISANCA3572H    FA      0    0   46 BRFACA1234T    BA  ---++-
0800  751   24  3.2    28.4   79    0    0   8    7     0     0  12 ++--+-

ISANCA0470T   PSINCA1157T    CF      0    0  436                    ------
0800 6094    0  0.5    29.3   81    0    0   7    7     0     0  12 ++++++

ISANCA0470T   ACRVCAXF94KI   FA      0    0  108                    ------
0800 1921    2  0.5    36.6   75    0    0   8    8     0     0  12 -----+

GRDNCA0386T   LOMTCA11CGO    CF      0    0   60                    ------
0800  740    3  0.5    32.2   82    0    0   6    6     0     0  12 +-----

SKOKCA0589T   SNDBCA0787T    FC      0  408    0                    ------
0800 3330    0  0.5    27.2   76 3232    0   8    8     0     0  12 ------

SKOKCA0589T   THCKCAXF49J    FA      0    0   52                    ---+-+
0800  910   35  3.8    29.8   83    0    0   8    8     0     0  12 +-+-++

SKOKCA0589T   PDRYCAXF30KI   FA      0    0  144                    ------
0800 2972    8  0.5    30.1   84    0    0  10   10     0     0  12 ---++-

ANHMCA0211T   CRIMCA1175S    FA      0    0  205                    ------
0800 3210    0  0.5    28.7   80    0    0   7    7     0     0  12 ---++-
```

Important! This report example is not intended to be an exact duplicate of the report provided by NTM. It is an example report that can be customized to the user's needs.

Table

Table 6 explains the headings in Figure 17.

Table 6 Description: Daily No Circuit Report

LABEL	MEANING
FROM	From Office name
TO	To Office name
SUF	Trunk group suffix
1WIN	Number of 1-way incoming trunk group
2WAY	Number of 2-way trunk groups
1WO	Number of 1-way outgoing trunk groups

Table 6 Description: Daily No Circuit Report

LABEL	MEANING
ALT	Name of alternate trunk group
NWT	Current hour and minute
PC	Peg count
OFL	Overflow
%OFL	Percentage of overflow
CCS/TRK	Number of CCS trunks
%OCC	Percentage of occupancy
IPC	Incoming peg count
ICCH	Incoming connections per circuit per hour
ACH	Attempts per circuit per hour
OCCH	Outgoing connections per circuit per hour
RRATT	Number of reroute attempts
RRSUC	Number of successful reroute attempts
PER COL	Number of periods data collected
5-MIN OFL	5-minute overflow period indicator
-	OK (no overflow)
+	Overflow (exceeded 0)
?	No data, error

This entry produces a Daily No Circuit Report for the twelve 5-minute periods starting at 14:00 of the day the command is produced.

> **Important!** The data required to produce the report in this example is not present in the database before 15:00. Therefore, do not execute the command before this time.

Examples

The following is an example of an `ncrpt` command entry:

```
$  ncrpt  from=all hour=16:00 pofl=0.5 ccs/trk=27.0 pocc=75
```

References

See Chapter 4, “User Report Writer” in the *Data Tables Guide* for more information on modifying example reports.

□

oprpt

Description

The `oprpt` command produces the Office Performance Report, which lists office performance data for a specific “From Office” and hour.

Syntax

```
oprpt from=office [hour=hour]
```

Parameters

`office`	”From Office” Name
`hour`	Start time of the hour being requested. The format for entering the hour is: YY/MM/DD-HH:mm

Important! Specify the hour on an even 1-hour boundary, for example, 13:00. If you do not specify an hour, the preceding current network hour is used.

System responses

Figure 18 shows the responses for the `oprpt` command.

Figure 18 oprpt command responses

IF the command is ...	THEN the system responds with ...
executed successfully	REPORT WRITER - - reporting by file '/nm/reports/oprpt_exe' Sending output to file '/usr/tmp/*tempfile*' Processing query. ... Retrieving data from Database. ... Executing report. ... Report '/nm/reports/oprpt_exe' completed The report can be found in /musr/admin/oprpt

When the command completes, the output is appended to the *“output_oprpt”* file in the “/musr/admin” directory. Use the *Linux* system `pg` command to view the file's contents. Remember to clean out this file periodically.

If you add the entry shown in Figure 19 to crontab, an Office Performance Report is produced each Monday through Friday for the hour beginning at 13:00.

Figure 19 Entry used with crontab to produce Office Performance Report

```
5 14 * * 1-5 ./etc/nm.env; /nm/cmdbin/oprpt shok
```

You can change this entry to produce a modified version of the report (assuming there is a modified darpt_exe report data file in the directory <directory name>) as shown in Figure 20.

Figure 20 Entry used with crontab to produce Modified Office Performance Report

```
5 14 * * 1-5 . /etc/nm.env; cd <directory name>;
   /nm/cmdbin/oprpt shok
```

Figure 21 shows a report generated by the `oprpt` command.

Figure 21 Office Performance Report

```
HOURLY OFFICE PERFORMANCE DATA REPORT:  NTM
DATE:                                   27-AUG-1994

OFFICE PERFORMANCE DATA           NWT:  1400
                                                                    PAGE    1

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 ESS4 OFFICE                                 DOM    INT
                 CANT   SKIP   MAN    MAN    CODE   CODE   RR     RR
                 GSC    GSC    SKIP   CANT   CTRL   CTRL   RCVD   FAIL
sim4e13ent01       440    430    480   470    450    460    530    540

10 DATA PERIODS
                 800    800    800    800    CANT   SKIP   CANT   SKIP
                 ATMPTS 6DIGS  10DIGS OTHER   ACC    ACC    CR     CR
                   490    500    510    520   390    400    410    420

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 ESS5 OFFICE                   TOTAL CALLS
                 ----------------------------------------  TOT    LINE
                 ORIG   INC    OUTG   TERM   INTRA  TAND   LOAD   BUSY
npvlda06e5h      1045   836     123    743      87    23   5423     95

10 DATA PERIODS     MATCHING LOSS
                 -------------------  CODE   TG     ATT    FAIL   MAN
                 ORIG   TERM   TAND   CAN    CAN    RR     RR     SKIP
                   83     12     34      0     0    123       8      0
```

Important! This report example is not intended to be an exact duplicate of the report provided by NTM. It is an example report that can be customized to the users needs.

Table

Table 7 explains the headings in Figure 21.

Table 7 Headers in Office Performance Report (Sheet 1 of 2)

FIELD NAME	DESCRIPTION
CANT GSC	Calls canceled by GSC (Group Signaling Congestion) cancel
SKIP GSC	Calls affected by GSC skip
MAN SKIP	Manual skip controls
MAN CANT	Manual CANT controls
CODE CTRL	Code controls
RR RCVD	Reroutes received
RR FAIL	Reroutes failed
800 ATMPTS	800 number attempts
800 6DIGS	Calls blocked by system 800 6-digit code controls
800 10DIGS	Calls blocked by system 800 10-digit code controls
800 OTHER	Calls blocked by system 800 for other reasons
CANT ACC	Calls canceled by ACC (dynamic overload control)
SKIP ACC	Calls affected by ACC-initiated skip
CANT CR	Calls canceled by CR
SKIP CR	Calls canceled by CR-initiated skip
TOTAL CALLS	Total number of calls
ORIG	Total originating calls
INC	Total incoming calls
OUTG	Total outgoing calls
TERM	Total terminating calls
INTRA	Total intrasystem calls
TAND	Total tandem calls
TOT LOAD	Total load
LINE BUSY	Total calls to line busy
ORIG	Originating matching loss

Table 7 Headers in Office Performance Report (Sheet 2 of 2)

FIELD NAME	DESCRIPTION
TERM	Terminating matching loss
TAND	Tandem matching loss
CODE CAN	Calls canceled by domestic code controls
TG CAN	Calls canceled by manual trunk group controls
ATT RR	Calls attempted to reroute by manual trunk group controls
FAIL RR	Calls failed by reroute by manual trunk group controls
MAN SKIP	Calls skipped by manual trunk group controls

Examples

The following is an example of an `oprpt` command entry:

```
$  oprpt  from=shokca0263t hour=14:00
```

References

See Chapter 4, “User Report Writer” in the *Data Tables Guide* for more information on modifying reports.

□

perfrep

Description

The perfrep (performance report) command outputs measurement data on the performance of the NTM application.

This measurement data can be output as a daily report, a summary report, or a page report. These reports may be output in either user- or machine-readable format. Feature 130, "Capacity and Usage Reporting" provides an enhanced version of these reports.

Syntax

```
perfrep [report=report] [date=date] [starttime=starttime]
   [endtime=endtime] [format=format] [-h]
```

Restrictions

Name-defined format must be used; parameter names are not optional.

```
Example: perfrep report=daily date=3/20
```

Nonexample: `perf rep daily 3/20`

Report type descriptions

A brief definition of the three report types follows:

1. The `daily` report (by default) prints data for the previous day. This consists of the NTM data for the 24-hour period.
2. The `summary` report (by default) prints the data for the current day. The data on the summary report is identical to the data on the daily report. Also, the starttime and endtime keywords can be used with the summary report and cannot be used with the daily report. By specifying starttime and endtime with the summary report, you can obtain a report for a range of hours during a particular day.

Parameters

`report` Type of report. Valid values are:

- daily
- summary

`date` output report for specified month and day. Enter date in format `mm/dd`. The default is the previous day for the daily report, or the current day for the summary or page reports. Reports can be output only if the data needed to produce the report is contained in a daily logfile (maximum 7 days) in the "/musr/meas/log" directory.

`starttime` starting hour of the report. Valid values are 0–23. The default is 0 hours.

Important! Not valid with `report=daily`

`endtime` ending hour of the report. Valid values are 0–23.

Notes:

1. Must be different from the starttime and is not valid with `report=daily`.
2. If no `starttime` or `endtime` is specified, then the default values are 0 and 23 respectively. If `starttime` is specified and `endtime` is not, then `endtime` gives up to an 8-hour report. If `endtime` is specified and `starttime` is not specified, then `starttime` gives up to an 8-hour report.

`format` Specifies the format of the report data.

- user = output the data in a report format for printing
- data = output the data in a format for reading into a spreadsheet

Important! The `data` option outputs the data in a format for downloading into a PC spreadsheet package and generating graphics, bar charts, etc.

Reference: For more information on this option, see “Viewing reports” (p. 4) in the *System Administration Guide*

`-h` Provides usage help, for example:

```
$ perfrep -h
usage: perfrep [report=daily|summary|page] [date={1-12}/{1-31}] [starttime={0-23}] [endtime={0-23}]]
[format=user|data] [-h]
```

System responses

Figure 22 shows a sample section of the responses for the `perfrep` command. The actual output contains multiple pages and is dependent on the type of report requested (daily, summary, or page).

Important! Only Trunk Groups that have a threshold-index value greater than 0 and are scheduled to collect data are reported in perfrep.

Figure 22 perfrep command output where 'report=daily'

```
Page  1
                        NTM Performance Report
                Short-Format Daily Report for cbnmga on 01/16/96

description             00:00  01:00  02:00  03:00  04:00  05:00  06:00  07:00
----------------------- ------ ------ ------ ------ ------ ------ ------ -----
Network Elements
  in Database

  DCCs activated            2      2      2      2      2      2      2      2
  switches activated       35     35     35     35     35     35     35     35
   reporting on-time        32     32     32     32     32     32     32     32
   reporting late           3      3      3      3      3      3      3      3
   not responding           0      0      0      0      0      0      0      0
  TGs reporting data       32     32     32     32     32     32     32     32

  num TG calcs (in K)       1      1      1      1      1      1      1      1

  TGs in exception         18     18     18     18     18     18     18     18
  Mach. in exception       29     29     29     29     29     29     29     29
  TTO exceptions           32     32     32     32     32     32     32     32

  Sec until excp upd       91     92     96     96     96     96     96     96
  Sec until excp EOP      158    160    164    164    165    164    163    163

LEGEND: # - data is missing             * - data is suspect
                                                                    Page  2
                        NTM Performance Report
                Short-Format Daily Report for cbnmga on 01/16/96


description             08:00  09:00  10:00  11:00  12:00  13:00  14:00  15:00
----------------------- ------ ------ ------ ------ ------ ------ ------ -----
Network Elements
  in Database
        .
        .
        .
        .
```

The output data files consist of a data item followed by a data key. The key is one of three characters.

IF the data key is ...	**THEN ...**
(blank space)	the data is good
#	the data is missing
*	the collection process is initialized or is missing a timed entry

Important! If the key is either * or #, then the data is suspect.

Examples

```
perfrep
  report=daily date=3/26 format=user
perfrep
  report=page date=1/31 starttime=1 endtime=10 format=data
perfrep
  report=summary date=3/26 starttime=1 endtime=10 format=user
```

References

Chapter 7, “Administrative Performance Reports” in the *System Administration Guide*

□

urwformat

Description

The urwformat (user report writer format) command is a filter designed to be used with the demand command in conjunction with the TUXEDO transaction system report writer. It has no arguments, and it accepts its input from standard input. The reformatted output is written to standard output.

The input must be in the form of a binary fielded buffer produced in accordance with the TUXEDO FML (Field Manipulation Language). The NTM demand command produces output in this form.

This output contains a single header containing a description of each of the records followed by the records themselves. The output records are sorted in the order in which they appeared in the SQL SELECT clause.

Syntax

```
urwformat
```

Restrictions

When generating new reports or porting old reports to new releases of NTM, do not use double quotes (") in .let, .if, and .print statements. Double quotes are not accepted in TUXEDO reports in these statements. Use single quotes (’) instead.

System responses

If the input contained unreadable FML records, the message “Query resulted in no records” appears.

Examples

Figure 23 shows the urwformat command used with the demand command inside a report source file.

Figure 23 urwformat command used with demand in report source file

```
.exec echo “SELECT to_office suffix period pc ofl
 %ofl %occ tg_rr_att sum_len FROM tgdat tgctl
 WHERE tg_srv = fi and office = grdn and period =
 11:55 and sum_len = 12” | demand | urwformat
```

Although it is shown in Figure 23 on four lines, the .exec command must be on one continuous line within the report source file.

In the retrieved data, each record meets the following restrictions:

- Each retrieved record would contain the nine fields listed in the select clause.
- All of these fields are trunk group reference (tgdat) and/or trunk group control (tgctl) data types.
- Only those records that have a trunk group server type of final (fi), an office ID of grdn, and a period of 11:55 (today) or earlier would be retrieved.
- The last restriction (sum_len = 12) causes each record to consist of summed data for the last 12 periods, ending with the period of 11:55-12:00.

References

See the `demand` command (5-21) for more information on SQL.

See Chapter 3, "SQL Interpreter" in the *Data Tables Guide*.

See the TUXEDO *FML Programmer's Guide* for detailed information on the Field Manipulation Language.

Related command(s)

"demand" (p. 21)
"fmltoasc" (p. 25)
"ongoing" (p. 51)

□

7 System Administration Commands

Overview

Purpose

This chapter lists and describes the NTM system administration commands. These commands are used on the host machine to perform administrative tasks.

Contents

This chapter discusses the following topics:

□

Command functions

Table

Table 1 lists each NTM system administration command and its function.

Table 1 System administration commands and functions

Command	Function
`act`	Activates data collection for an office, switch, or DCC (Data Collection Concentrator)
`act_dcc`	Activates all the end offices defined for a DCC office
`addntmuser`	The addntmuser command adds s user's RDB credentials.
`deact`	Deactivates data collection for the specified office(s) or DCC
`deact_dcc`	Deactivates all end offices associated with a DCC
`delntmuser`	The delntmuser command removes s user's RDB credentials.
`exptrules`	Allows enabling/disabling of rules processing of exceptions
`snw_admin`	Sets up and administers subnetworks
`start_arc`	Starts an ARC process
`startnet`	Starts the Ethernet on the computer along with associated services
`stop_arc`	Stops an ARC process
`stopnet`	Gracefully shuts down the Ethernet on the computer
`stopsys`	Stops or shuts down the system
`viperm`	Command to safe editing of permissions file.

□

act

Description

The act (activate) command activates data collection for one or more offices, switches, or DCCs (Data Collection Concentrators).

Syntax

```
act office=office [type=type]
```

Before you begin

Before using the act command on a *4ESS* switch, use the linkstat command (9-9) to verify that the network element you are activating has “conn” displayed in the Session column.

To activate any office for data collection that is connected through a DCC, activate both the switch and its corresponding DCC. For current DCC status, see the linkstat command (9-9).

Parameters

You can have multiple values for each of these parameters. The values are separated by the plus sign (+)

office	NTM office, DCC name, NMS office or all
type	One of the following function types:

- meas
- discrete
- control
- audit
- csv
- all (default is all)

System responses

Figure 1 shows the responses for the `act` command.

Figure 1 act system responses

IF the command is …	THEN the system responds with …
executed successfully	IP (In Progress) followed by *office* activated for *types(s)*; where *office* is the user-defined office and *type* is either meas, discrete, control, or audit
executed successfully, but the office was previously activated	*office* already activated for *type(s)*
executed, but the switch is not associated with a DCC and data collector	No DCC/data collector associated with the entity. NTM database and shared memory updated. This response indicates there is no active association between this switch and a DCC or data collector. The activation status has been updated as requested, and it will be used when the switch is associated with a connected DCC.
not executed successfully because there is a problem communicating with the data collector. For example, the system is stopped.	cannot communicate with data collector, NTM DB updated This response indicates the database has been changed. When the system is restarted, the requested activation type(s) are used.
not executed successfully	the appropriate error message

Examples

This example shows office “anhmca0512t” activated for all function types.

```
$ act office=anhmca0512t
```

References

Chapter 9, “Adding and Removing Network Elements” in the *System Administration Guide*

Related command(s)

“deact” (p. 11)
“linkstat” (p. 9)

□

act_dcc

Description

Use the `act_dcc` (activate data collection concentrator) command to activate all the end offices defined under a given DCC office. This command can be used to activate all end offices for a specified DCC office for which this host is primary or to activate all end offices for which this host is in takeover mode.

> **Important!** The `act_dcc` command will activate only the end offices that it knows about at the time.

Syntax

`act_dcc` *`dcc`*

Parameters

`dcc` The name of the data collection concentrator and its associated end offices that you wish to activate.

System responses

Figure 2 shows sample output for the `act_dcc` command.

Figure 2 act_dcc Command Output

```
$ act_dcc dccA
  IP

dccA activated for audit discrete control meas
  IP

office1 activated for audit discrete control meas

office2 activated for audit discrete control meas

office3 activated for audit discrete control meas

office4 activated for audit discrete control meas

office5 activated for audit discrete control meas
  IP

office6 activated for audit discrete control meas

office7 activated for audit discrete control meas
```

```
office8 activated for audit discrete control meas
```

Examples

```
$  act_dcc  dccA
```

References

For information about BDR (Backup and Disaster Recovery) procedures, see Chapter 12, “BDR Administration on a Host” in the *System Administration Guide*.

Related command(s)

“act” (p. 4)
“deact_dcc” (p. 13)
“linkstat” (p. 9)

□

addntmuser

Description

The `addntmuser` command adds a new user's RDB credentials.

Syntax

```
addntmuser username=username
```

Parameters

`username` The `addntmuser` command may be supplied a single NTM application username on the command line or, if no username is given on the command line, you will be prompted to supply one. The username must match an operating system username.

> **Important!** An NTM application username must be added before the user is given access to NTM application data.

Passwords / logins

You must be `nmadm` to run this command.

System responses

If a username argument is not supplied on the command line to `addntmuser` then the user will be prompted as follows to enter a username:

```
Enter a valid username:
```

If an invalid username is entered the following will appear and execution is halted:

```
addntmuser failed - <username> is not a valid OS level
   username
```

Upon success the following will be displayed:

```
Creating RDB credentials for <username> ()...
**
**
**
**
Grant succeeded.
Grant succeeded.
Grant succeeded.
Grant succeeded.
RDB credentials for <username> created
```

References

Chapter 2, "Adding and Removing Users on the Host" in the *System Administration Guide*

Related command(s)

“delntmuser” (p. 15)

□

deact

Description

The `deact` (deactivate) command deactivates data collection for the specified network elements.

Syntax

```
deact office=office [type=type]
```

Parameters

`office`	NTM office, DCC name, NMS office or all
`type`	Function type:

- meas
- discrete
- control
- audit
- csv
- all (default is all)

Notes

1. You can enter multiple values for *`office`* and *`type`* separated by the plus sign (+).
2. Function type `meas` refers to the collection of report data.

System responses

Figure 3 shows the responses for the `deact` command.

Figure 3 deact system responses

IF the command is ...	THEN the system responds with ...
executed successfully	IP followed by: *`office`* deactivated for *`type`*(s)
executed successfully but the office was already deactivated for that type	*`office`* already deactivated for *`type`*(s)
executed, but the switch is not associated with a DCC and data collector	No DCC/data collector associated with the entity. NTM database and shared memory updated. This response indicates there is no active association between this switch and a DCC or data collector. The activation status has been updated as requested, and it will be used when the switch is associated with a connected DCC.

Figure 3 deact system responses

IF the command is …	THEN the system responds with …
not executed successfully because there is a problem communicating with the data collector (for example, the system is stopped).	cannot communicate with data collector, NTM DB updated This response indicates the database has been changed. When the system is restarted, the requested activation *type*(s) will not be activated for data collection.
not executed successfully	the appropriate error message

Examples

The following example shows office “abcd” deactivated for all function types:

```
$ deact office=abcd
```

The following example shows office “atlanta05” deactivated for all function types:

```
$ deact office=atlanta05
```

Related command(s)

“act” (p. 4)
“linkstat” (p. 9)

□

deact_dcc

Description

Use the deact_dcc (deactivate data collection concentrator) command to deactivate all the end offices defined under a given DCC office. This command can be used to deactivate all end offices for a specified DCC office for which this host is primary or to deactivate all end offices for which this host is in takeover mode.

> **Important!** The deact_dcc command will deactivate only the end offices it knows about at the time.

Syntax

```
deact_dcc dcc
```

Parameters

dcc The name of the DCC and its associated end offices that you wish to deactivate

System responses

Figure 4 shows sample output for the deact_dcc command.

Figure 4 deact_dcc command output

```
$ deact_dcc dccA
  IP

office1 deactivated for audit discrete control meas

office2 deactivated for audit discrete control meas

office3 deactivated for audit discrete control meas

office4 deactivated for audit discrete control meas

office5 deactivated for audit discrete control meas
  IP

office6 deactivated for audit discrete control meas

office7 deactivated for audit discrete control meas

office8 deactivated for audit discrete control meas

  IP
```

```
dccA deactivated for audit discrete control meas
```

Examples

The following is an example of a `deact_dcc` command entry:

```
$  deact_dcc dccA
```

References

For more information about BDR procedures, see Chapter 12, “BDR Administration on a Host” in the *System Administration Guide*.

Related command(s)

“act_dcc” (p. 7)
“deact” (p. 11)
“linkstat” (p. 9)

□

delntmuser

Description

The delntmuser command removes user’s RDB credentials.

Syntax

```
delntmuser username=username
```

Parameters

`username` The `delntmuser` command may be supplied a single NTM application username on the command line or, if no username is given on the command line, you will be prompted to supply one.

> **Important!** The given username RDB credentials will be removed regardless of whether or not the user’s OS level ID still exists or not.

Passwords / logins

You must be the `nmadm` to run this command.

System responses

If a username argument is not supplied on the command line to `delntmuser` then the user will be prompted as follows to enter a username:

```
Enter a valid username:
```

If an invalid username is entered the following will appear and execution is halted:

```
delntmuser failed - <username> is not a valid username
Valid usernames are:
<list of valid usernames>
```

Upon success the following will be displayed:

```
RDB credentials for <username> removed
```

References

Chapter 2, “Adding and Removing Users on the Host” in the *System Administration Guide*

Related Command(s)

“addntmuser” (p. 9)

□

exptrules

Description

The `exptrules` (exception rules) command allows rules processing of exceptions to be enabled/disabled.

Syntax

```
exptrules act=action
```

Parameters

`action` Valid values are:

- off — disable rules processing of exceptions
- on — enable rules processing of exceptions
- stat — report if rules processing is enabled or disabled

System responses

Figure 5 shows the responses for the `exptrules` command.

Figure 5 exptrules system responses

IF the action specified is ...	THEN the system responds with ...
off	IP followed by the system prompt
on	IP followed by the system prompt
stat	IP followed by "Rules processing is *<enabled | disabled>*"

Examples

Following are examples of the `exptrules` command.

```
exptrules on
exptrules off
exptrules stat
exptrules act=on
exptrules act=off
exptrules act=stat
```

References

See the "Filter File" (p. 25) in the *Record Base Administration Guide*.

□

snw_admin

Description

The snw_admin (subnetwork administration) command can be used to set up and administer subnetworks and user groups.

After you enter the snw_admin command you are provided with a display of current subnetworks, valid user groups and their permissions for each subnetwork, and a menu of administrative options.

The menu of options allows you to add or delete subnetworks and user groups associated with subnetworks. You can also assign subnetwork and command permissions as well as modify these existing permissions.

> **Important!** Adding a user group for subnetwork association requires the group to exist prior to adding. Contact your System Administrator for needed user group creation.

> **Important!** The snw_admin command supports up to 30 characters length groups.

Syntax

```
snw_admin
```

Passwords / logins

You must be logged in as nmadm to execute this command.

System responses

Figure 6 is an example of an snw_admin command display.

Figure 6 snw_admin Command Output

```
09-07-92   13:15:41       SUBNETWORK ADMINISTRATION
--------------------------------------------------------------------------


   SUBNETWORK    TYPE                          USER GROUPS
  ------------  -----       ---------------------------------------------
      NMC         M             NM      RB     USR
      SUB1        R             SUB1NM
      SUB2        R             SUB2NM



--------------------------------------------------------------------------
```

```
 MAIN MENU

1. CREATE A NEW SUBNETWORK        5. MODIFY NAME/PERMISSIONS
2. DELETE AN EXISTING SUBNETWORK  6. VIEW SUBNETWORKS
3. ADD USER GROUP                 7. QUIT
4. DELETE USER GROUP


 MENU OPTION:
```

References

Chapter 11, “Subnetwork Administration” in the *System Administration Guide*

Related command(s)

“snw_info” (p. 22)

□

start_arc

Description

The start_arc command starts an ARC process. ARC provides a way for external systems to perform demand-like queries in a much more efficient way than using the demand command. The response time is shorter because the ARC process is able to start and process many demand-type queries without restarting. Since the initialization is performed only once, each subsequent query is handled faster than it would be through the demand command because less processing is required.

The queries are not handled from the command line. Instead, a TCP/IP socket number is returned when an ARC process is started. All SQL queries coming into that socket are processed, and the responses are sent back out over the same socket. The ARC process is intended to replace the demand and fmltoasc commands for external systems.

Syntax

```
start_arc [-print option] [-f <strg>] [-m <strg>] [-s <char>] [-n
   <char>] [-c <char>] [-i <strg>] [-t <n>]
```

Feature restrictions

This command is available only with the purchase of Feature 106, “Active Request Controller”.

Intended users

This command is not intended as a user command. It is to be used only by external systems that intend to use the ARC interface for accessing data.

Parameters

This command takes the form of a standard *Linux* command with the following options:

print option Valid values are:

- p = Print the data in the form <fieldname>=<value>.
- r = Print only the retrieved fields, that is, those from the select clause of the SQL file (see “SQL command syntax” (p. 23)). The default is to print all the fields contained within the fielded buffer.
- e = Print the value and exception level for calculated fields. The default is to print only the value.
- l = Print the label associated with the select clause. The label is optional in the SQL file, therefore this option may display a null string.

`-f <strg>` Use the string <strg> as the field delimiter string. The default delimiter string is the blank.

Important! Do not use the `-f` option with the `-w` and/or `-n` options because the `-f` option takes precedence.

`-m <strg>` Use the string <strg> as the message delimiter string. The default delimiter string is a new line.

`-s <char>` Use the character <char> as the subfield delimiter character. The default delimiter character is a blank. This option is used when a field is made up of several subfields.

`-n <char>` Use the character <char> as the delimiter character before a number. The default delimiter character is a blank.

`-c <char>` Use the character <char> as the delimiter between fields in calculated fields. The default delimiter character is a blank.

`-i <strg>` Use the string <strg> as the invalid data string. The default invalid data string is "-1".

`-t <n>` Use the number <n> as the time format definition, where:

- n = 1 *Linux* system date format
- n = 2 hour and minute
- n = 3 date, hour, and minute
- n = 4 hour, minute, and second (default)
- n = 5 date, hour, minute, and second

System responses

If the command is executed successfully, the system responds with the process ID of the started ARC process and the socket number to which it is attached. A return code of 0 indicates success.

If the command is not executed successfully, the system responds with an appropriate error message. A return code of 1 indicates failure.

References

For more information on ARC, see Chapter 13, "ARC Administration" in the *System Administration Guide*.

See the *TUXEDO FML Programmer's Guide* for detailed information on FML.

See Chapter 1, "All Data Fields" in the *Data Tables Guide* for a list of valid field ID names and values.

Related command(s)

“demand” (p. 21)
“fmltoasc” (p. 25)
“stop_arc” (p. 25)

□

startnet

Description

The `startnet` (start Ethernet) command allows you to start the Ethernet on the computer by enabling the Ethernet protocol and ARP (Address Resolution Protocol) over the network interface.

The `startnet` command gives you the list of lan cards to select from. You can select any card to start the Ethernet.

Syntax

```
startnet
```

Passwords / logins

You must be logged in as `root` to run the `startnet` command.

Before you begin

If the machine has more than one lan card before executing this command, identify the lan card to start the Ethernet.

Feature restrictions

If you have BDR or you are collecting data through DCCs, you must start the NTM system with the `startsys` command after running the `startnet` command.

Related command(s)

"stopnet" (p. 27)

□

startsys

Description

The startsys (start system) command allows you to start the NTM application system. If the Oracle software is not running, it will be started when the startsys command is executed.

Syntax

```
startsys
```

Passwords / logins

If you are logged in with any login other than nmadm, you will be required to enter a password to execute this command.

If you are logged in as nmadm, no password is required.

If you have purchased Feature 22, “NMADM Login Accountability” you will not be prompted for the nmadm password.

System responses

Figure 7 shows the responses for the startsys command.

Figure 7 startsys system responses

IF the command is …	THEN the system responds with …
executed successfully	$ startsys 8920 Network Traffic Managment Software Copyright (c) 1990-2010. Alcatel-Lucent. Oracle Listener already started Oracle already started 8920 NTM Relational DB is starting up The 8920 NTM software is now starting (will be ready in about 2 minutes).

Important! Your password is not visible on the terminal screen.

References

For more information on using the startsys command and for an example of command output, see Chapter 4, “Starting and Stopping the System” in the *System Administration Guide*.

Related command(s)

“stopsys” (p. 28)

□

stop_arc

Description

The stop_arc command stops a particular ARC process.

This command is not intended as a user command. It is to be used only by external systems that use the ARC interface for accessing data.

Syntax

```
stop_arc  proc
```

Feature restrictions

This command is available only with the purchase of Feature 106, “Active Request Controller”.

Intended users

Only the external system that started the ARC process or superuser may use this command.

Parameters

proc	The process ID of an ARC process to be stopped.

System responses

If the command is executed successfully, no message is printed and a return code of 0 is given.

If the command is not executed successfully, the system responds with an appropriate error message and a return code of 1.

Examples

The following is an example of a stop_arc command to stop an ARC process with a process ID of 1234.

```
$  stop_arc  1234
```

References

Chapter 13, “ARC Administration” in the *System Administration Guide*

Related command(s)

“demand” (p. 21)
“fmltoasc” (p. 25)
“start_arc” (p. 19)

□

stopnet

Description

The `stopnet` (stop Ethernet) command allows you to gracefully shut down the Ethernet on the computer by disabling the Ethernet protocol and ARP over the network interface. The `stopnet` and `startnet` commands should be used to clear Ethernet problems.

The `stopnet` command gives you the list of lan cards to select from. You can select any card to start the Ethernet.

Syntax

```
stopnet
```

Passwords / logins

You must be logged in as `root` to run the `stopnet` command.

Before you begin

If the machine has more than one lan card before executing this command, identify the lan card to start the Ethernet.

Feature restrictions

If you have BDR or you are collecting data through DCCs, you must start the NTM system with the `startsys` command after stopping and starting the Ethernet with `stopnet` and `startnet`.

Related command(s)

"startnet" (p. 22)

□

stopsys

Description

The stopsys (stop system) command allows you to stop the NTM application system. Executing the stopsys command will not stop the Oracle software.

Syntax

```
stopsys
```

Passwords / logins

If you are logged in with any login other than nmadm, you will be required to enter a password to execute this command.

If you are logged in as nmadm, no password is required.

If you have purchased Feature 22, "NMADM Login Accountability" you will not be prompted for the nmadm password.

Interdependencies

All audit processes will be terminated when the stopsys command is executed.

System responses

Figure 8 shows the responses for the stopsys command.

Figure 8 stopsys system responses

IF the command is ...	THEN the system responds with ...
executed successfully	$ stopsys Shutting down admin processes ... Server Id = 0 Group Id = 1 Machine = rp01: server or resource in-u. 0 processes stopped. The 8920 NTM software is now stopped Continuing with 8920 NTM Relational DB module shutdown Beginning shut-down of autonomous processes 07/05 09:20:00 SHUT-DOWN complete.

Important! Your password is not visible on the terminal screen.

References

For more information on using the `stopsys` command and for an example of command output, see Chapter 4, “Starting and Stopping the System” in the *System Administration Guide*.

Related command(s)

“startsys” (p. 23)

□

viperm

Description

The purpose of `viperm` command is to edit in a safe way the */nm/etc/permissions* file. This command uses the EDITOR environment variable to determine a text editor. If this variable is not set, vim is assumed.

Syntax

```
viperm [-c][-f <file>]
```

Passwords / logins

If you are logged in with any login other than `nmadm`, you will be required to enter a password to execute this command.

If you are logged in as `nmadm`, no password is required.

If you have purchased Feature 22, “NMADM Login Accountability” you will not be prompted for the `nmadm` password.

Parameters

`-c`	Checks the syntax of previously edited permissions file without opening it in the editor.
`-f <file>`	Changes the default permissions file location used by the command to the temporary one provided by <file> parameter.

System responses

Figure 9 shows the responses for the `viperm` command.

Figure 9 viperm system responses

IF the command is ...	THEN the system responds with ...

executed successfully	`[nmadm@lnx22 ~]$ viperm`
	`[nmadm@lnx22 ~]$`

executed successfully with `-c` parameter	`[nmadm@lnx22 ~]$ viperm -c`
	`Permissions syntax OK.`

Figure 9 viperm system responses

IF the command is ...	THEN the system responds with ...

executed without `-c` parameter and the syntax of introduced changes is wrong	`[nmadm@lnx22 ~]$ viperm -f ./permissions`
	`ERROR in line 1: `acta`. There must be exactly 3 tokens separated by `|` or none.`
	`Wrong permissions syntax.`
	`Hit enter to continue editing or ctrl-c to discard changes and quit.`

executed with `-c` parameter and the syntax of introduced changes is wrong	`[nmadm@lnx22 ~]$ viperm -c -f ./permissions`
	`ERROR in line 1: `a`. There must be exactly 3 tokens separated by `|` or none.`
	`Wrong permissions syntax.`

References

For more information about */nm/etc/permissions* file, see Chapter 3, “Permissions file” in the *System Administration Guide*.

□

8 UDDM/UDNEI Commands

Overview

Purpose

This chapter lists and describes the NTM UDDM/UDNEI commands. These commands are used to manage User Defined Data Modeling and User Defined Network Element Interfaces tasks.

Reference: For using the commands, see "UDDM/UDNEI Administration" (p. 1) in the *System Administration Guide*.

Important! You must be logged as `nmadm` or some other user in the snm group to run the UDDM/UDNEI commands (See: NMADM login accountability).

Contents

This chapter discusses the following topics:

□

Command functions

Table

Table 1 lists each NTM UDDM/UDNEI command and its function.

Table 1 System administration commands and functions

Command	Function
`encrypt_param`	Encrypts specific parameter in the instance switch file.
`install_dcol`	Installs new UDNEI dcol.
`manage_audit`	Manages audits.
`manage_control`	Manages controls.
`manage_ccrules`	Populates CC configuration tables.
`manage_dcol_params`	Manages data collection parameter files.
`manage_instance_params`	Manages parameter files.
`manage_ssh`	Manages ssh keys.
`manage_uddm`	Manages UDDM data types.
`manage_udneitype`	Manages UDNEIType files.
`udnei_act`	Start data collectors for specific UDNEIType.
`udnei_deact`	Stop data collectors for specific UDNEIType.
`manage_aggr`	Manage periodic data aggregation engine.
`manage_stat`	Manages statystical thresholding engine.

□

encrypt_param

Description

The `encrypt_param` command allows you to encrypt specific parameter in the *"/musr/rb/udnei/<dcol>/<UDNEIType>/<clliname>"* file. In case of encrypting several parameters, the `encrypt_param` command must be executed for each parameter separately. After encryption, this file is still editable and later is used with the `manage_instance_params` command.

Syntax

```
encrypt_param file name value
```

Parameters

`file`	Full or relative path to the <*clliname*> file.
`name`	Specific parameter name which exist in the "*/musr/rb/udnei/<dcol>/<UDNEIType>/<clliname>*" file.
`value`	Value of the parameter which must be encrypted.

System responses

If given <*clliname*> file does not exist the system responses the following:

```
<clliname> file does not exist.
```

Result

After successful execution of the `encrypt_param` command, encrypted *`value`* of the parameter *`name`* will be appended at the end of <clliname> *`file`*.

Examples

The following is an example of a `encrypt_param` command entry:

```
$ encrypt_param /musr/rb/udnei/my_dcol/my_nei/ne1 password
  x5rty84r
```

References

"Provisioning a network element instance" (p. 16) in the *System Administration Guide*

Related command(s)

"manage_dcol_params" (p. 23)

"manage_instance_params" (p. 25)

install_dcol

Description

Use the install_dcol command to add a new dcol to the system. This command copies the executable provided by the user to *"/nm/udbin/"* and creates a directory called *"/musr/rb/udnei/<dcol>"*.

Syntax

```
install_dcol dcol_name
```

Parameters

dcol_name	Full or relative path to the new dcol file.

System responses

The following table shows the responses for the install_dcol command.

IF the command is ...	THEN the system responds with ...
executed successfully	DCOL successfully installed.
executed, but <*dcol*> binary does not exist.	<dcol> binary does not exist.
executed, but could not create a <dcol> directory.	Could not create */musr/rb/udnei/<dcol> directory*.

Result

After successful execution of the install_dcol command the dcol executable is copied to *"/nm/udbin/"* folder. This command creates the *"/musr/rb/udnei/<dcol>"* folder. If specific <dcol> exists the command prompts user and if agreed rewrites and creates a <dcol> of the same name.

Examples

The following is an example of a install_dcol command entry:

```
$ install_dcol /my_directory/my_dcol
```

References

"Using an existing UDNEI Data Collector" (p. 11) in the *System Administration Guide*

□

Keys File

Purpose

The */musr/uddm/keys/key_vals* file is used to provide Choice definitions and Range definitions which can be referenced in the *<UDDMType>_ref* and *<controlname>_control* files. The `manage_uddm` command always updates the database with the key values if they are changed. Before editing the *key_vals* file prepare a backup copy of that file. A common error can occur when the database already contains the reference for the keys and the user is trying to update them. You have to restore the copy of the *key_vals* file or remove the references in that file.

Keys File

This is the example of */musr/uddm/keys/key_vals* file:

```
$ cat key_vals
# ChoiceValue
#generic, choiceName, choiceType, value;
rsm5_0,limittype_enum,enum,Critical+System+Endpoint+Subnet+Realm+Unknown
   +Sip;

# RangeValue
#generic, rangeName, min, max
#rsm5_0,range_nextone_limit,0,10000;
rsm5_0,range_nextone_ulograte,0,3600;
```

Keys File Parameters

Below is a list of parameters in the */musr/uddm/keys/key_vals* file.

Parameters for the ChoiceValue section:

`generic` Generic Name (`all` for all generics). For UDNE switches you have to provide real generic.

`choiceName` Name of the Choice. The value must adhere to Oracle column naming rules. Those rules are not enumerated here.

`choiceType` Type of Choice. The allowed values are:

- enum - numerical value,
- query - defines a query which creates the Choice.

`value` Allowed values:

- enum - numeric values separated by the plus (+) sign.
- query - SQL statement returning values of one attribute.

Parameters for the RangeValue section:

`generic` Generic Name (*`all`* for all generics). For UDNE switches you have to provide real generic.

`rangeName` Range name. The value must adhere to Oracle column naming rules. Those rules are not enumerated here.

`min` Minimum value.

`max` Maximum value.

Related Command(s)

“manage_control” (p. 10), “manage_uddm” (p. 30)

□

manage_audit

Description

Use `manage_audit` command to move the definition of audits to the database. The definition of audits must be written in the *"/musr/uddm/audits/<audit_name>_audit"* file. Then you can use audits by using command `udaudit`.

Syntax

```
manage_audit act auditname
```

Parameters

`act` Parameter which defines specific action. You can put one of the following:

- add - adding new audit,
- del - removing audit,
- mod - modifying audit,
- test - testing audit,
- assign - assign the audit to the specific dcol according to the parameters in the */musr/uddm/audits/assignments/audit_assignments* file.

`auditname` Audit name. This parameter is case sensitive.

Result

After successful execution of the `manage_audit` command the definition of audit is written into the database.

Template File

The following is the template for the *"/musr/uddm/audits/<audit_name>_audit"* file. Figure 1 shows the example of that file.

```
#AuditDef
#Generic,AuditName,AuditType,AuditDest,HelpText
```

Template Parameters

Below is a list of the parameters used in the *"/musr/uddm/audits/<audit_name>_audit"* file:

`Generic` generic name

`AuditName` Name of the audit. It must be the same as prefix of the filename.

`AuditType` Audit type. It can be one of the following:

- list - audit type for lists.
- control - audit type for controls.
- wsdl - audit type for collecting the definition of webservices from Network Element.

`AuditDest` Name of audit destination:

- for list audits - name of the reference table
- for control audits - name of the control definition

`HelpText` help comment (up to 4000 characters).

Template Example

Figure 1 Example of the *"/musr/uddm/audits/ratelimit_audit"* **file.**

```
#AuditDef
#Generic,AuditName,AuditType,AuditDest,HelpText
rsm5_0,ratelimit,control,RATELIMIT_CTL,Manual ratelimit control audit;
```

Assignments File Example

The following is an example of the *musr/uddm/audits/assignments/audit_assignments* file:

```
ratelimit,nextonersm,soap;

eplist,nextonersm,soap;

rlmlist,nextonersm,soap;
```

Examples

The following is an example of a `manage_audit` command entry:

```
$ manage_audit add ratelimit
```

References

"Managing Audits" (p. 22) in the *System Administration Guide*

Related command(s)

"manage_uddm" (p. 30)

□

manage_control

Description

After using `manage_uddm` command for creating controls the appropriate table and file */musr/uddm/controls/<UDDMType>_control.template* are created. You can manage controls by using `manage_control` command. You have to edit the file */musr/uddm/controls/<UDDMType>_control.template* and save it as */musr/uddm/controls/<UDDMType>_control* file.

Syntax

```
manage_control act controlname
```

Parameters

`act` Parameter which defines specific action. You can put one of the following:

- add - adding new control,
- del - removing control,
- mod - modifying control,
- test - testing control,
- assign - assign the control to the specific dcol according to the parameters in the */musr/uddm/controls/assignments/control_assignments* file.

`controlname` Control name. This parameter is case sensitive.

Result

After successful execution of the `manage_control` command the definition of control is written into the database.

Template File

The following is the template for the *"/musr/uddm/controls/<UDDMType>_control.template"* file. Figure 2 shows the example of that file.

```
#ControlDef
#Generic,ControlName,Action,OnModNew,UDDMTable,HelpText

#CtrlParam
#Generic,ParamName,Type,Required,Default,DefaultValue,TableName,FieldNam
   e,ChoiceName,RangeName,HelpText
```

Template Parameters

Below is a list of the parameters used in the *"/musr/uddm/controls/<UDDMType>_control"* file. For the ControlDef section:

`Generic` Generic name

`ControlName` Name of the control. It must be the same as prefix of the filename.

`Action` Available action: add, mod, del.

`OnModNew` Allowed values; y, n. If set to "y" this control is written as new and old one is moved to history.

`UDDMTable` Table name for collecting controls.

`HelpText` help comment (up to 4000 characters).

Parameters for the CtrlParam section:

`Generic` Generic name

`ParamName` Parameter name.

`Type` Parameter type. Allowed values; string, double, integer.

`Required` Allowed values: y, n.

`Default` Set control as default. Allowed values: y, n.

`DefaultValue` Default Value, only if default set to "y".

`TableName` Corresponding Table Name.

`FieldName` Corresponding Field Name.

`ChoiceName` If this value is not set the NTM is getting ChoiceName from the Table definition.

`RangeName` If this value is not set the NTM is getting RangeName from the Table definition.

`Helptext` Help comment (up to 4000 characters).

Template Example

Figure 2 Example of the ***"/musr/uddm/controls/ratelimit_control"*** **file.**

```
#ControlDef
#Generic,ControlName,Action,OnModNew,UDDMTable,HelpText
rsm5_0,ratelimit,mod,y,ratelimit_ctl,;

#CtrlParam
#Generic,ParamName,Type,Required,Default,DefaultValue,TableName,FieldNam
e,ChoiceName,RangeName,HelpText
```

```
rsm5_0,devicename,String,y,n,,ratelimit_ctl,DeviceName,,,Device Name;
rsm5_0,partition,String,y,n,,ratelimit_ctl,Partition,,,NexTone partition
name; rsm5_0,bucket,String,y,n,,ratelimit_ctl,Bucket,,,NexTone queue
name; rsm5_0,devicemode,String,y,n,,ratelimit_ctl,DeviceMode,,,NexTone
mode; rsm5_0,layer,String,y,n,,ratelimit_ctl,Layer,,,NexTone layer;
rsm5_0,limittype,String,y,n,,ratelimit_ctl,LimitType,,,Limit Type;
rsm5_0,limitrate,Double,n,n,,ratelimit_ctl,LimitRate,,,Steady-state rate
limit; rsm5_0,limitburst,Double,n,n,,ratelimit_ctl,LimitBurst,,,Burst
rate limit; rsm5_0,ulograte,Double,n,n,,ratelimit_ctl,UlogRate,,,Ulog
rate limit;
rsm5_0,threshold,Double,n,n,,ratelimit_ctl,Threshold,,,Threshold;
```

Assignments File

The following is an example of the *musr/uddm/controls/assignments/control_assignments* file:

```
ratelimit,nextonersm,soap;
```

Examples

The following is an example of a `manage_control` command entry:

```
$ manage_control add widgetcontrol
```

References

"Managing Controls" (p. 23) in the *System Administration Guide*

Related command(s)

"manage_uddm" (p. 30)

□

manage_ccrules

Description

The manage_ccrules command:

- creates/removes necessary directory structure,
- validates rules template file against Oracle schema and rule semantics,
- populates the CC configuration tables.

If you populate an analysis job you do not have to provide the *UDNEIType* and *dcol*. The last three parameters in manage_ccrules command are optional.

Before you begin

Before attempting to use the manage_ccrules command be sure that the "*<UDDMtype>_cc.template*" file is properly prepared (See "Setting up CC rules" (p. 14) in the *System Administration Guide*).

Syntax

```
manage_ccrules act UDDMType [UDNEIType] [dcol] [suffix]
```

Parameters

act Parameter which defines specific action. You can put one of the following:

- add - adding new ccrules,
- del - removing rules,
- mod - modifying the rules,
- test - testing rules.

UDDMType Full UDDMType name.

UDNEIType Full UDNEIType name (up to 30 char.).

dcol List of full dcol names, separated by plus sign (+).

suffix Unique instance number. Use it when you want to create more CC instances for specific UDDMType and UDNEIType pair.

System responses

The following table shows sample output for the manage_ccrules command.

IF the command is ...	THEN the system responds with ...
executed, but other instance of manage_ccrules is running.	manage_ccrules is already running.

IF the command is ...	THEN the system responds with ...
executed with *act=add* and CC rules for given combination of *<udneitype> <uddmtype> <suffix>* exists.	`You cannot perform add operation on existing sourcename.`
Specific cc rule is not defined in single quotes.	`CC rule for field <fieldname> must be defined in quotes ‘’.`

Result

After successful execution of the `manage_ccrules` command the CC rules from the *<UDDMType>_cc.template* file are inserted into the Oracle Database. This command also creates *“/musr/hod/<UDNEIType>/<UDDMType>[_suffix]”* directory. The udnei data collectors will place the collected data into this directory for CC to consume.

Important! As CC process those data files, the files are moved to appropriate sub-directories: processed, error, and garbage. The system will automatically clean these sub-directories over time.

Examples

The following are the examples of a `manage_ccrules` command entry:

```
$ manage_ccrules add WIDGET my_UDNEI my_dcol 1
$ manage_ccrules del WIDGET my_UDNEI my_dcol 1
```

Template File

The following is the template for the *“/musr/uddm/ccrules/<source_name>_cc.template”* file where *source_name* is a combination of <UDDMType>_<UDNEIType>_<suffix>. Figure 3 shows the example of that file.

```
# Source Config - DO NOT REMOVE/CHANGE THIS LINE!!
# casesensitive ;
<database/temporary field indicator>;

# Source Config - numfield
<numfield value>;

# Lookups - DO NOT REMOVE/CHANGE THIS LINE!!
# name , table , keys , output ;
# Rules - DO NOT REMOVE/CHANGE THIS LINE!!
# target , dbfield , logic ;
<field name>,<database/temporary field indicator>,'<cc rule>';
# Apply threshold tests - DO NOT REMOVE/CHANGE THIS LINE!!
```

Template Parameters

Below is a list of the parameters used in the *“/musr/uddm/ccrules/<source_name>_cc.template”* file:

`casesensitive`
This parameter specifies if data values should be treated as ‘case-sensitive’. Enter `y` for case-sensitive.

`numfield` The number of fields in the CSV file. CC validates input records against this value. Any record that does not have this number of CSV fields exactly is discarded.

`name` Name of the lookup object.

`table` The table containing the mapping data (usually the <UDDMType>_ref table).

`keys` The keys of the lookup (specified in order), separated by colons.

`output` The columns that can be the return of the lookup, separated by colons.

`target_field` Name of field (populated in <UDDMType>_dat table or virtual field for calculations)

`dbfield` Indicates that field given by *`target_field`* is a field in <UDDMType>_dat table or virtual field for calculations. Allowed values are ‘y’ or ‘n’.

`expression` CC rule (up to 1000 characters). It must be defined in quotas.

Template Example

Figure 3 Example of the *“/musr/uddm/ccrules/<source_name>_cc”* file.

```
# Source Config - DO NOT REMOVE/CHANGE THIS LINE!!
# casesensitive ;
y;

#Source Config - numfield
1;

# Lookups - DO NOT REMOVE/CHANGE THIS LINE!!
# name, table, keys, output;
# Rules - DO NOT REMOVE/CHANGE THIS LINE!!
# target , dbfield , logic;
period, y, 'field(1)';
office, y, 'field(2)';
raw1, y, 'field(3)';
raw2, y, 'field(4)';
calc1, y, 'divide(convfield(raw1),convfield(raw2))';
calc1_suspect, y, ‘choose(compare(convfield(calc1),”100”,
   GREATER),”TRUE”,”s”,”v”)’;
```

```
# Apply threshold tests - DO NOT REMOVE/CHANGE THIS LINE!!
```

Important! Note that the "# Apply threshold tests" directive is required. In the event that a CC rule for a field requires the result of threshold testing, place that field and its primitive below this directive. For example, the end_time field in the code_event analysis job depends on the value of the MAX_EXC_LV field. As such, it is placed below the "#apply threshold tests" directive.

References

"Setting up CC rules" (p. 14) in the *System Administration Guide*

Related command(s)

"manage_uddm" (p. 30)

"manage_aggr" (p. 46)

□

Primitives

Overview

The following is the list of primitives used for creating rule sets for Configurable Converter in NTM.

- add
- and
- choose
- compare
- convfield
- divide
- field
- formatdatetime
- getsubstring
- isequal
- lookup
- mod
- multiply
- not
- null
- or
- subtract
- user

add

```
add(arg1,arg2,...,argn)
```

`arg1 - argn` Rule type. Rules that return numeric value.

Executes all rules (*`arg1 - argn`*). The results are float values which are summed to calculate return value.

and

`and(arg1,arg2,...,argn)`

`arg1 - argn` Rule type. Rules that returns string value.

Executes all rules (*arg1 - argn*) consecutively up to first rule that returns string which does not begin with ”T” character. If such a rule is found `and()` returns string "FALSE", otherwise returns "TRUE".

choose

`choose(expr,val1,result1,...,valn,resultn,[default])`

`expr` Rule type.

`val1 ... valn`Rule type.

`result1 .. resultn`Rule type.

`default` Rule type.

The primitive executes *expr* rule in order to obtain base string. Then executes each *var* argument until result of *var* rule is lexicographically equal to base string. If match is found `choose()` executes corresponding *result* rule and immediately returns its result. Otherwise executes *default* rule if specified and return its result. If *default* is omitted NULL is returned.

compare

`compare(lval, rval, oper)`

`lval` Rule type.

`rval` Rule type.

`oper` String type.

The primitive executes *lval* and *rval* rules to retrieve left and right side values respectively. Both values are converted to floating point numbers and compared according to *oper* argument. If conversion is possible and comparison is true `compare()` returns string "TRUE". Otherwise returns string "FALSE". Possible values for oper argument are:

- "greater"
- "less"
- "equal"
- "greater_equal"
- "less_equal"

convfield

`convfield(fieldname)`

`fieldname` String type. Name of previously stored field.

Returns value of previously calculated field under given name. The field can be both DB field stored in the periodic table or virtual field. Define a rule for given `fieldname`, that will be calculated before first call to `convfield(fieldname)`. Otherwise `convfield(fieldname)` will not compile preventing CC instance to run.

divide

```
divide(numerator,denominator)
```

`numerator` Rule type. Rule that returns numeric value.

`denominator` Rule type. Rule that returns numeric non zero value.

Executes `numerator` and `denominator` rules. If `denominator` result value is NULL or 0 `divide()` returns NULL. Otherwise calculates and returns value of division: `numerator` value / `denominator` value.

field

```
field(fieldid)
```

`fieldid` String type. Field ID in input file.

Retrieves value of given field in CC input data record. The `fieldid` is the position of the field in CSV file.

formatdatetime

```
formatdatetime(datetime,datefmt,timefmt,timezone)
formatdatetime(datetime,timefmt,timezone)
```

`datetime` Rule type. field() or convfield() rule, that returns date/time stamp.

`datefmt` String type. Date pattern.

`timefmt` String type. Time pattern.

`timezone` String type. String "YES" or rule which returns time zone.

This primitive can be used to convert a wide array of date/time formats. The `datetime` argument must be one of convfield() or field() primitive, which is executed to gain date/time stamp. The result is parsed according to `timefmt` and optionally `datefmt` patterns. The `datefmt` could contain tokens:

- %m - month number
- %d - day of month
- %j - day of year
- %c - four-digit year
- %y - two-digit year

The `timefmt` could contain tokens:

- %h - hour

- %m - minutes
- %s - seconds

If the timezone is string "YES" or this is a rule that return time zone of the element that generated the time stamp, time stamp normalization is performed.

getsubstring

```
getsubstring(srcstr,startpos,[length])
getsubstring(srcstr,regex)
```

`srcstr` Rule type. Evaluates to source string.

`startpos` Rule type. Evaluates to numerical representation of character offset.

`length` Rule type. Evaluates to numerical representation of substring length.

`regex` String type. Regular expression pattern.

Executes *srcstr* rule and threat its value as source string. If second argument is of type Rule executes *startpos* and optionally *length* to retrieve numeric values of character offset and substring length respectively. Then extracts a substring out of the value returned by *srcstr*, which starts at given offset up to *length* characters. If *length* argument is omitted all characters from offset till end of source string are extracted. Use `user()` primitive to provide fixed values of offset and length. If second argument is of type String, then is threated as regular expression pattern, which is applied on source string. Then first matching substring is returned. In case of any error `getsubstring()` returns NULL.

isequal

```
isequal(field1,field2,caseSense)
```

`field1` Rule type.

`field2` Rule type.

`caseSense` String type. Case sensitive flag.

Executes *field1* and *field2* rules and compare resulting values. If both results are NULL or lexicographically equal, `isequal()` returns string "TRUE". Otherwise string "FALSE" is returned. If *caseSense* argument is "YES", comparison is not case sensitive.

lookup

```
lookup(lkup_name,key1,key2,...,keyn,out_fld)
```

`lkup_name` String type. Defined lookup name.

`key1 - keyn` Rule type. Rule which result with value of key.

`out_fld` String type. Name of output field.

Execute all rules from *key1* to *keyn*. If at least one of those rules returns NULL `lookup()` primitive also returns NULL. Results of executed rules are used as values of key fields in defined lookup of given *lkup_name*. Using calculated keys a search is performed in defined lookup in order to retrieve and return value of field *out_fld*.

> **Important!** `lookup()` primitive requires correctly defined lookup table. Such definition consist of:
>
> - name - Name of lookup same as 'lkup_name'
> - table - Name of source table in data base
> - keys - List of key fields name in source table
> - output - List of fields name in source table available for retrieval

Example:

Lookup definition:

```
# Lookups
# name , table , keys , output ;
my_ref_lkup, widget_ref, office+farend, office_name+office_loc;
```

Lookup use:

```
# Rules
# target , dbfield , logic ;
office, y, field(1);
farend, y, field(2);
office_name, y,
   lookup(my_ref_lkup,convfield(office),convfield(farend),office_name);
   office_loc, y,
   lookup(my_ref_lkup,convfield(office),convfield(farend),office_loc);
```

mod

`mod(numerator/denominator)`

`numerator` Rule type. Rule that returns numeric value.

`denominator` Rule type. Rule that returns numeric non zero value.

Executes *numerator* and *denominator* rules. If one of results is NULL or cannot be converted to numeric value or *denominator* value is 0 `mod()` returns NULL. Otherwise `mod()` calculates and returns remainder of division: *numerator* value / *denominator* value.

multiply

`multiply(arg1,arg2,...,argn)`

`arg1 - argn` Rule type. Rules that return numeric value.

Executes all given rules (*arg1 - argn*). The not NULL results are converted to float values. Then calculates and returns product of retrieved values.

not

`not(rule)`

`rule` Rule type. Rule that returns boolean string value.

Executes *`rule`* argument. If result of *`rule`* is string that begins with “T” character, `not()` returns "FALSE". If result of *`rule`* is string that begins with “F” character, `not()` returns "TRUE". Otherwise `not()` returns NULL.

null

`null()`

Always returns NULL

or

`or(arg1,arg2,...,argn)`

`arg1 - argn` Rule type. Rules that returns boolean value.

Executes all given rules (*`arg1 - argn`*) consecutively up to first rule that returns string which begins with “T” character. If such a rule is found `or()` returns string "TRUE", otherwise returns "FALSE".

subtract

`subtract(arg1,arg2,...,argn)`

`arg1 - argn` Rule type. Rules that return numeric value.

Executes all given rules (*`arg1 - argn`*). The not NULL results are converted to float values. Then subtract from first retrieved value rest of retrieved values. Returns result of calculation.

user

`user(string)`

`string` String type. User defined string.

Always returns given *`string`*. Used to pass user defined value as Rule argument.

□

manage_dcol_params

Description

The manage_dcol_params command allows you to load or reload the data collection parameters.

Syntax

```
manage_dcol_params act dcol
```

Parameters

act Parameter which defines specific action. You can put one of the following:

- add - adding the dcol parameters to the database,
- del - removing the dcol parameters from the database,
- mod - modifying the dcol parameters stored in the database.

dcol Full dcol name.

System responses

When you enter the manage_dcol_params command and other instance of manage_dcol_params is running then system returns the following response:

```
manage_dcol_params is already running
```

Result

After successful execution of the manage_dcol_params command the dcol parameters are inserted into the Oracle Database.

Examples

The following are the examples of a manage_dcol_params command entry:

```
$ manage_dcol_params add my_dcol
$ manage_dcol_params del my_dcol
```

Template File

The following is the template for the *"/musr/rb/udnei/<dcol>/parameters"* file. Figure 4 shows the example of that file.

```
paramName=<param name>,paramType=<param type>,optional=<optional>,
   default=<defaultvalue>,validation=<validation>;
```

Template Parameters

Below is a list of the parameters used in the *"/musr/rb/udnei/<dcol>/parameters"* file:

`paramName` up to 30 characters.

`paramType` The allowed values are:

- string - string up to 256 characters.
- number - integer number.
- URI - IP address in dotted decimal notation or DNS name and port number (typically port number is 161), includes udp/tcp protocol designations.
- SQL - SQL query executed in the NTM database.

`optional` indicates if this is an optional parameter. The allowed values are: true, or false.

`defaultvalue` The default value for the instance parameter.

`validation` Range, or allowed values of the parameter used for validation:

- For number parameters range contains min value and max value separated by the“-” sign.
- For string value provide the list of allowed values separated by the “|” sign.

Template Example

Figure 4 Example of the ***“/musr/rb/udnei/<dcol>/parameters”*** **file.**

```
paramName=IP,paramType=URI,optional=false
paramName=User,paramType=string,optional=false
paramName=Passwd,paramType=string,optional=false
paramName=Periodicity,paramType=number,default=5
paramName=Offset,paramType=number,optional=false,validation=0-240000
paramName=Filename,paramType=string,optional=false
paramName=Directory,paramType=string,optional=false
paramName=Timeout,paramType=number,optional=true
paramName=SNMPversion,paramType=string,default=V2C,validation=V1|V2C|V3
paramName=Links,paramType=SQL,optional=true
```

References

“Using an existing UDNEI Data Collector” (p. 11) in the *System Administration Guide*.

Related command(s)

“install_dcol” (p. 5)

manage_instance_params

Description

The `manage_instance_params` command loads or reloads the contents of the parameter file from the *"/musr/rb/udnei/<dcol>/<UDNEIType>/"* directory. The parameters in the *"/musr/rb/udnei/<dcol>/<UDNEIType>/<clliname>"* file depend on the specific data collector and its parameters. The `manage_instance_params` is executed by the `create office` command and then all the parameters are added into the Oracle Database. After `create office`, you can run the `manage_instance_params` command only with a *`mod`* flag to modify parameters.

Syntax

```
manage_instance_params act clliname
```

Parameters

`act` Parameter which defines specific action. You can put one of the following:

- add - adding the parameters to the database,
- del - removing the parameters from the database,
- mod - modifying the parameters stored in the database.

`clliname` Office name. You can enter multiple values separated by the plus sign (+).

System responses

The following table shows the system responses for the `manage_instance_params` command.

IF the command is ...	THEN the system responds with ...
executed, but another instance of `manage_instance_params` is running.	manage_instance_params is already running.
executed with *`act=add/mod`*, but parameters for this *`<clliname>`* are exists.	Parameters for <clliname> already exists.
executed successfully.	Parameters for <clliname> were added/modified/deleted.

Result

After successful execution of the `manage_instance_params` command parameters for every dcol used by <clliname> office are added, removed or modified in Oracle Database.

Examples

The following is an example of a `manage_instance_params` command entry:

```
$ manage_instance_params mod ne1
```

Template File

The following is the template for the *“/musr/rb/udnei/<dcol>/<UDNEIType>/<clliname>”* file. Figure 5 shows the example of that file.

```
paramName=<param name>,paramValue=<param value>;encrypted=<y/n>
```

Template Parameters

Below is a list of the parameters used in the *“/musr/rb/udnei/<dcol>/<UDNEIType>/<clliname>”* file:

`paramName`	up to 30 characters.
`paramValue`	up to 256 characters.
`encrypted`	allowed values are 'y' or 'n'.

For string value provide the list of allowed values separated by the “|” sign.**Template Example**

Figure 5 Example of the ***“/musr/rb/udnei/<dcol>/<UDNEIType>/<clliname>”*** **file.**

```
paramName=IP,paramValue=ftp://sol450adev.cb.lucent.com:21,encrypted=n
paramName=User,paramValue=ftpuser,encrypted=n
paramName=Passwd,paramValue=p{1#qqq,encrypted=y
paramName=Periodicity,paramValue=15,encrypted=n
paramName=Offset,paramValue=0,encrypted=n
paramName=Filename,paramValue=sn05-pp.csv,encrypted=n
paramName=Directory,paramValue=/home/ftpuser,encrypted=n
paramName=Timeout,paramValue=1000,encrypted=n
paramName=Links,paramValue=SELECT name FROM Links WHERE eid=<clliname>
```

Related command(s)

“create” (p. 5)

□

manage_ssh

Description

The manage_ssh command manages the SSH configuration. This command is utilizing office parameters added to the NTM database by the manage_instance_params command. The manage_ssh is executed by the manage_instance_params command for the SSH offices. The manage_ssh tests an SSH connection between NTM host and the network element. If succeeded, it exchanges SSH keys between NTM and the remote network element.

Syntax

```
manage_ssh [-f] act clliname
manage_ssh act type filename [passphrase] [comment]
manage_ssh --help
```

Parameters

act — Parameter which defines specific action. You can put one of the following:

- testconn - testing an ssh connection,
- copykeys - exchanging keys between NTM and the network element,
- genkeys - generating SSH keys on the NTM host.

clliname — Office name. You can enter multiple values separated by the plus sign (+).

type — Algorithm used to generate the keys. Allowed values are DSA and RSA.

filename — Name of the newly generated SSH private key.

passphrase — Optional. Passphrase to encode SSH private key.

comment — Optional. Used at the end of the SSH public key.

--help — Displays help.

-f — Force flag, executes command without confirmation from the user.

System responses

The following table shows the system responses for the manage_ssh command.

IF the command is ...	THEN the system responds with ...
executed, but other instance of manage_ssh is running.	manage_ssh is already running .

IF the command is ...	THEN the system responds with ...
executed, but Feature 440, “UDNEI SSH Support” is disabled.	Feature not active exit
executed with the `-f` option, but the attempt to establish the connection has failed.	Please verify office parameters and run 'manage_ssh testconn | copykeys " + entName + "' manually. OR Host is not available. Host name: XXX
excuted with the `testconn` option, but either the remote host or your own ssh authentication keys were changed and have not been exchanged.	Your keys are either out of date or have not been installed on the remote host. Execute 'manage_ssh copykeys' to solve the problem.
executed successfully.	Connection with the office established. Office name: <clliname> OR Keys copied successfully for the office:<clliname> OR Keys successfully created.

Result

After successful execution of the `manage_ssh command,` the result of the particular *`act`* parameters are following:

- testconn - remote element’s public key will be added to the *known_hosts* file on the NTM
- copykeys - NTM’s public key will be added to the *auth_hosts* file on the remote element
- genkeys - new pair of SSH keys will be generated

Examples

The following is an example of a `manage_ssh` command entry:

```
$ manage_ssh testconn ne1
$ manage_ssh copykeys ne1
$ manage_ssh genkeys RSA /musr/nmadm/.ssh/id_rsa ‘’
   ‘nmadm@hostname’
```

Related command(s)

“manage_instance_params” (p. 25)

□

manage_uddm

Description

For managing UDDM data types, use the `manage_uddm` command. This command allows you to create new datatypes, remove, and modify existing datatypes. The files: *<UDDMType>_dat, <UDDMType>_thr,* and *<UDDMType>_ref, <UDDMType>_ctl* for defining the meta data for a new datatype and controls must be stored in the *"/musr/uddm/tables"* directory. The periodic data schema file (*<UDDMType>_dat*) is required. The threshold (*<UDDMType>_thr*) and reference (*<UDDMType>_ref*) files are optional. `Manage_uddm` command provides validation of these files and creates table names:

- <UDDMType>_dat,
- <UDDMType>_ref,
- <UDDMType>_thr,
- <UDDMType>_ctl.

Creation of <UDDMType>_dat and <UDDMType>_ctl tables is mutually exclusive for the same UDDMType.

This command is also used to create database tables that contain standalone or switch-independent reference data, which is available for use by all UDDM types. The standalone reference data table requires a <RefDataTye>_ref file, which is created using the same template as the <UDDMType>_ref file.

> **Important!** Default table layouts and alerts for UDDM types are automatically replicated on BDR hosts. It causes that table layouts or alerts for particular datatype can be populated on hosts on which this datatype was not added by manage_uddm command. However information about this table layouts and alerts is not available on the GUI. If `manage_uddm` returns this information user is prompted whether adding datatype should be continued or not. Continuing this process may cause that new default table layouts and alerts will be populated across all BDR hosts. You must ensure that this datatype is synchronized across the affected hosts.

Syntax

```
manage_uddm act UDDMType [subnets=subnets] [keys=keys]

manage_uddm -option

manage_uddm act RefDataType
```

Parameters

`act` This parameter defines a specific action for UDDM types or reference data types. Valid values include:

- add - creating UDDMType/RefDataType in the system,
- del - removing UDDMType/RefDataType from the system,
- mod - recreating UDDMType/RefDataType in the system (remove then create),
- test - testing status of tables and files for the specific UDDMType/RefDataType.

`UDDMType` UDDMType name. This parameter is case insensitive.

`subnets` Allowed values: “y”, “n”. If set to “y” `manage_uddm` command creates subnetwork fields.

`keys` Allowed values: “y”, “n”. If set to “y” `manage_uddm` command updates key values from the *‘/musr/uddm/keys/key_vals”* file

`option` Allowed values:

- -j - `manage_uddm` setting up joins provided in the *“/musr/uddm/tables/common”* file.
- -k - `manage_uddm` updates key values from the *‘/musr/uddm/keys/key_vals”* file.
- -a - `manage_uddm` defines atypical days used by statistical thresholding engine. Atypical days are defined in */musr/uddm/tables/atypicaldays* file.

`RefDataType` Name of a type of standalone reference data. This is used to create a database table to hold standalone or switch-independent reference data (reference data that can be used by any UD type). This value must match the name of an existing */musr/uddm/tables/<RefDataType>_ref* file, which is constructed from the *template_ref file* in the same directory.

System responses

The following table shows the responses for the `manage_uddm` command.

IF the command is …	THEN the system responds with …
executed, but other instance of `manage_uddm` is running.	`manage_uddm is already running.`
executed with *`act=add`* and given *`<uddmtype>`* exists.	`You cannot perform add operation on existing uddmtype.`

IF the command is …	THEN the system responds with …
executed with *act=mod* or *act=del* and CC instance for given <*uddmtype*> is running.	`CC for <uddmtype> is running.`
executed, but `manage_uddm` finds errors in *<UDDMType>dat/_ref/_thr* file.	`<UDDMType>_dat/_ref/_thr file contains errors`

If user defines data model with the same name on BDR host and LDAP information was replicated. This information is not available on the GUI.	`A table layout and alert for this data type already exists (probably due to BDR).`
	`Continuing will cause the default table layout to be overwritten for all host.`
	`Please ensure the data type is synchronized across the affected hosts. Continue? (y/n)`

executed with *act=del* and for a given <*uddmtype*> the data aggregation exists.	`Aggregation exists for selected model. Use manage_aggr del <model> to delete aggregations and repeat the command.`

Sample Output

The following is manage_uddm sample output from adding a <UDDMType>:

```
$ manage_uddm add demo
Do you want to add subnetwork field for demo_dat table? (y/n)
y
Subnetwork field will be added for demo_dat table
  IP
Creating DEMO_DAT table

  IP
Creating DEMO_REF table

  IP
Creating DEMO_THR table

manage_uddm add successfully completed
```

Result

After successful execution of the `manage_uddm` command the tables for types given by UDDMtype are added, removed, modified in the Oracle Database. The files "*/musr/rb/uddm/<UDDMType>_ref.template*" file and *"/musr/uddm/ccrules/<UDDMType>_cc.template*" are created. When creating controls the file *"/musr/uddm/controls/<UDDMType>_control.template*" is created.

Example

The following are the examples of a `manage_uddm` command entry:

```
$ manage_uddm add WIDGET
$ manage_uddm del WIDGET
$ manage_uddm -j
$ manage_uddm add REFDATA
```

Template Files

To create the files: *<UDDMType>_dat, <UDDMType>_ref, <UDDMType>_thr,* and *<UDDMType>_ctl* you can use the *template_dat*, *template_ref, template_thr,* and *template_ctl* files located in the */musr/uddm/tables* directory. You must leave the comment lines started with "#", they are mandatory. To set up joins edit the *"/musr/uddm/tables/common"* file.

To create the file *<RefDataType>_ref* to define a standalone reference data table, use the *template_ref* template file.

The *template_dat* file supports the following entries:

```
#TableSchema/ChronInfo
#TableName, HelpText, Chronological, UpdatePeriod, UpdateEvent,
   TableGroup;

#FieldSchema/UserFieldSchema
#Fieldname, Label, FieldType, SuspectCol, DataType, Keyfield, HelpText;

#JoinTables - optional entry for setting up joins
#DataTable, RefTable, DataCols, RefCols, JoinClause, JoinType;
```

The *template_ref* file supports the following entries:

```
#TableSchema
#TableName, HelpText;

#FieldSchema/UserFieldSchema
#Fieldname, Label, FieldType, DataType, ChoiceName, Keyfield, HelpText;

#Relationship - optional table for setting up table relationships
#Linkset, Hyperlink, ReferringPage, ReferringTable, ReferringKey,
   ReferencedPage, ReferencedTable, ReferencedKey;
```

The file *template_thr* supports the following entries:

```
#TableSchema
#TableName, Help_text;

#FieldSchema/UserFieldSchema
#Fieldname, Label, FieldType, DataType, Keyfield, HelpText;
```

The file *template_ctl* supports the following entries:

```
#TableSchema
#TableName, HelpText, Chronological, updatePeriod, updateEvent

#FieldSchema/UserFieldSchema
#Fieldname, Label, FieldType, DataType, ChoiceName,RangeName,
   Keyfield,NotNull, HelpText;
```

The file *"/musr/uddm/tables/common"* supports the following entries:

```
#JoinTables - optional entry for setting up joins
#DataTable, RefTable, DataCols, RefCols, JoinClause, JoinType;
```

Template Parameters

<UDDMType>_dat file

Below is a list of the parameters used in the *"<UDDMType>_dat"* file. Section #JoinTables has common parameters with *"/musr/uddm/tables/common"* file:

Section #TableSchema/ChronInfo: This section must contain one record with six attributes.

`TableName` name of a chronological (i.e. periodic) table. The value must match the <UDDMType> portion of the <UDDMType>_dat file name. It is limited to 20 characters.

`HelpText` help comment (up to 4000 characters).

`Chronological`

allowed values are ‘y’ or ‘n’. This value is set to ‘y’ for the “<UDDMType>_dat” table and to ‘n’ for other tables.

`UpdatePeriod` period value in minutes. Allowed values are: 5m, 15m, 30m, 60m.

`UpdateEvent` Available values are: ‘exc’ or ‘netevent’. This parameter is `exc' if type is populated as a result of exception processing of data collected from network elements, and `netevent' if type is populated by an enhanced thresholding analysis job.

`TableGroup` This is an optional parameter used by the GUI on the Periodic Data Browser page to provide groupings for tables in the drop-down menu.

Section #FieldSchema/UserFieldSchema: This section must contain at least two records, and each record should contain seven attributes.

Important! In the #FieldSchema/UserFieldSchema section, there must be one and only one record in which the Keyfield attribute is set to "y" and the FieldType value is set to "nearend".

There must be at least one row where the FieldType attribute is set to either "raw" or "calc" and the Keyfield attribute is set to "n".

`Fieldname` name of a column in table given by Tablename (up to 20 characters). The value must adhere to Oracle column naming rules. Those rules are not enumerated here. (A very usual case would be names of fields that contain percentages. "% Success" is not be a valid field name since % is an illegal column name character in Oracle. A field name like p_success or pct_success should be chosen instead.)

`Label` an alternative name of a column (up to 30 characters). This label is used for user interface display purposes. (If a fieldname of "pct_success" was specified, the GUI column label on the Periodic Data Browser page can be set to "% success" is so desired.)

`FieldType` allowed values for the FieldType are:

- nearend – indicates, that column contains near end name (typically its office). One column must exist with this FieldType.
- farend – indicates that column contains far end name. One column can exist with this FieldType.
- ref - indicates that column contains reference data.
- time - indicates that column contains a timestamp.
- attr – indicates that column contains an attribute. This is generally string data, possibly part of the key, that is not subject to threshold testing.
- raw – indicates that column contains raw data. This is an indication to the system that this data may be subject to thresholding.
- calc – indicates that column contains calculated data. This is an indication to the system that this is a derived field and this field may be subject to thresholding.

`SuspectCol` indicates if the Oracle table should provide a column indicating suspect data for this field. This column is optional and is used to display a possible "?" inside the data value fields on the Periodic Data Browser, indicating that the data is questionable or suspect. Allowed values are:

- y – indicates that a suspect data column should be added for this field
- n – indicates a suspect data column should not be added for this field

- blank – uses the default value of “y”. When leaving this field blank, you must still enter a comma to indicate the field has been left blank (do not enter a space).

If a SuspectCol value is provided, a single character column will be created as part of the table schema. The suspect data column name will be set automatically to the fieldname with “_sp” appended. That field must be populated with a CC rule, setting its value to “v” if the associated data is valid and to “s” if the associated data is suspect.

`DataType` Oracle type. Standard Oracle datatype syntax (such as ‘number’, ‘date’, ‘varchar(n)’, ‘varchar2(n)’, where ‘n’ is an integer value specifying a length) is used in all cases except for numbers that have decimal places, which must be entered as `‘number(precision:scale)’` (you have to use the ‘:’ sign instead of ‘,’ for proper validation). . For more information, see the Oracle documentation.

`Keyfield` indicates if column given by Fieldname is a key in table given by Tablename. Allowed values are ‘y’ or ‘n’.

Note: Set Keyfield to “y” only if FieldType is one of these values: nearend, farend, attr, or ref.

`HelpText` help comment (up to 4000 characters). This data is displayed as field help on the Periodic Data Browser page.

Section #JoinTables:

`DataTable` name of a table. It must end with ‘_dat’ (up to 17 characters).

`RefTable` name of a reference table.

`DataCols` list of names of column in table given by DataTable separated with ‘|’ sign.

`RefCols` list of names of column in table given by RefTable separated with ‘|’ sign.

`JoinClause` SQL join statement. For more information, see Oracle Documentation.

`JoinType` type of join given by join_clause (for example: left, left outer). For more information, see Oracle documentation.

<UDDMType>_ref or <RefDataType>_ref file

Below is a list of the parameters used in the *“<UDDMType>_ref”* file (*<RefDataType>_ref* file uses the same parameters):

Section #TableSchema: This section must contain only one record with two attributes.

`TableName` name of a reference table. The value must match the <UDDMType> portion of the <UDDMType>_ref file name. It is limited to 20 characters.

`HelpText` help comment (up to 4000 characters).

Section #FieldSchema/UserFieldSchema: Each record in this section must contain seven attributes.

> **Important!** In the #FieldSchema/UserFieldSchema section, there must be one and only one record in which the Keyfield attribute is set to “y” and the FieldType attribute is set to “nearend”.

`FieldName` name of a column in table given by TableName (up to 20 characters). The value must adhere to Oracle column naming rules. Those rules are not enumerated here. (A very usual case would be names of fields that contain percentages. “% Success” is not be a valid field name since % is an illegal column name character in Oracle. A field name like p_success or pct_success should be chosen instead.)

`Label` alternative name of a column (up to 30 characters).

`FieldType` allowed values for the FieldType are:

- nearend – indicates, that column contains near end name (typically its office). One column must exist with this FieldType.
- farend – indicates that column contains far end name. One column can exist with this FieldType.
- attr – indicates that column contains an attribute.
- ref - indicates that column contains reference data.
- time - indicates that column contains a timestamp.

> **Important!** When creating a *<RefDataType>_ref* file, the only FieldType allowed is ref.

`DataType` Oracle type. Standard Oracle datatype syntax (such as ‘number’, ‘date’, ‘varchar(n)’, ‘varchar2(n)’, where ‘n’ is an integer value specifying a length) is used in all cases except for numbers that have decimal places, which must be entered as `‘number(precision:scale)’` (you have to use the ‘:’ sign instead of ‘,’ for proper validation). . For more information, see the Oracle documentation.

`ChoiceName` name of choice for column given by FieldName (up to 30 characters).

`Keyfield` indicates if column given by FieldName is a key in table given by Tablename. Allowed values are ‘y’ or‘n’.

`HelpText` help comment (up to 4000 characters).

Section #Relationship:

`Linkset` identifier for a group of links.

`Hyperlink` indicates if relationship is a hyperlink. Allowed values are ‘y’ or ‘n’.

`ReferringPage`
name of a referring page.

`ReferringTable`
referring table name (up to 17 characters). This is the same as *`Tablename`* given in <UDDMType>_dat file.

`ReferringKey` name of the column given in <UDDMType>_dat file (it must be a key).

`ReferencedPage`
name of a referenced page.

`ReferencedTable`
referenced table name (up to 17 characters). It always the same as *`Tablename`* given in <UDDMType>_dat file.

`ReferencedKey`
name of the column given in <UDDMType>_dat file (it must be a key). The presence of the "<UDDMType>_thr" file indicates to the `mange_uddm` command that a thresholding rules table is required. The TableSchema entry must also be populated to provide the name of that table. The schema of that table is pre-defined; hence, the fields of this table need not be input by the user.

<UDDMType>_thr file

Below is a list of the parameters used in the *"<UDDMType>_thr"* file:

Section #TableSchema: This section must contain one and only one record with the following two attributes.

`TableName` name of a thresholding table. The value must match the <UDDMType> portion of the *<UDDMType>_thr* file name. It is limited to 20 characters

`HelpText` help comment (up to 4000 characters)

<UDDMType>_ctl file

Below is a list of the parameters used in the *"<UDDMType>_ctl"* file:

Section #TableSchema

`TableName` name of a table. The value must match the <UDDMType> portion of the *<UDDMType>_ctl* file name. It is limited to 20 characters

`HelpText` help comment (up to 4000 characters)

`Chronological`
allowed values are 'y' or 'n'. This value is set to 'y' for the "<UDDMType>_dat" table and to 'n' for other tables.

`UpdatePeriod` period value in minutes. Allowed values are: 5, 15, 30, 60.

`UpdateEvent` Available values are: ‘exc’ or ‘netevent’. This parameter is `exc' if type is populated as a result of exception processing of data collected from network elements, and `netevent' if type is populated by an enhanced thresholding analysis job.

Section #FieldSchema/UserFieldSchema:

`Fieldname` name of a column in table given by Tablename (up to 17 characters). The value must adhere to Oracle column naming rules. Those rules are not enumerated here. (A very usual case would be fields that are a percentage. “% Success” would not be a valid field name since % is an illegal column name character in Oracle. A field name like p_success or pct_success should be chosen instead.)

`Label` an alternative name of a column (up to 30 characters). This label is used for user interface display purposes. If a fieldname of “pct_success” was specified, the GUI column label on the Periodic Data Browser page can be set to “% success” is so desired.

`FieldType` allowed values for the FieldType are:

- nearend – indicates, that column contains near end name (typically its office). One column must exist with this FieldType.
- farend – indicates that column contains far end name. One column can exist with this FieldType.
- ref - indicates that column contains reference data.
- attr – indicates that column contains an attribute. This is generally string data, possibly part of the key, that is not subject to threshold testing.
- raw – indicates that column contains raw data. This is an indication to the system that this data may be subject to thresholding.
- calc – indicates that column contains calculated data. This is an indication to the system that this is a derived field and this field may be subject to thresholding.

`DataType` Oracle type. Standard Oracle datatype syntax (such as ‘number’, ‘date’, ‘varchar(n)’, ‘varchar2(n)’, where ‘n’ is an integer value specifying a length) is used in all cases except for numbers that have decimal places, which must be entered as `‘number(precision:scale)’` (you have to use the ‘:’ sign instead of ‘,’ for proper validation). . For more information, see the Oracle documentation.

`Keyfield` indicates if column given by Fieldname is a key in table given by Tablename. Allowed values are ‘y’ or ‘n’.

`NotNull` Indicates null parameter.

Template Examples

The following figures show the examples of the *“widget_dat”* (Figure 6), *“widget_ref”* (Figure 7), *“widget_thr”* (Figure 8), *“ratelimit_ctl”* (Figure 9), and *“common”* (Figure 10) files.

Figure 6 Example of the *“widget_dat”* file.

```
#TableSchema/ChronInfo
#TableName, HelpText, Chronological, UpdatePeriod, UpdateEvent,
   TableGroup;
WIDGET_DAT,Help of WIDGET_DAT table,y,5m,exc;

#FieldSchema/UserFieldSchema
#Fieldname, Label, FieldType, SuspectCol, DataType, Keyfield, HelpText;
office, NULL, nearend, NULL, varchar2(16), y, NULL;
raw1, NULL, raw, NULL, number, n, raw1 help;
raw2, NULL, raw, NULL, number, n, raw2 help;
calc1, NULL, calc, calc1_sp, number, n, calc1 help;

#JoinTables - optional entry for setting up joins
#DataTable,RefTable,DataCols,RefCols,JoinClause,JoinType
#WIDGETGUI_DAT,WIDGETGUI_REF,OFFICE,OFFICE,WIDGETGUI_DAT.OFFICE=WIDGETGU
   I_REF.OFFICE,outer;
```

Figure 7 Example of the *“widget_ref”* file.

```
#TableSchema
#TableName, HelpText;
WIDGET_REF,NULL;

#FieldSchema/UserFieldSchema
#Fieldname,Label,FieldType,DataType,ChoiceName,Keyfield,HelpText;
office,NULL,nearend,varchar2(16),NULL,y,NULL;
comm,NULL,ref,varchar(30),NULL,n,NULL;
test2_type,NULL,ref,varchar2(10),test1_type_ch,n,NULL;

#Relationship - optional table for setting up table relationships
#Linkset, Hyperlink, ReferringPage, ReferringTable, ReferringKey,
   ReferencedPage, ReferencedTable, ReferencedKey;
```

Figure 8 Example of the *“widget_thr”* file.

```
#TableSchema
#TableName, HelpText;
WIDGET_THR, NULL;

#FieldSchema/UserFieldSchema
#Fieldname, Label, FieldType, DataType, Keyfield, HelpText;
```

Figure 9 Example of the *“ratelimit_ctl”* file.

```
#TableSchema
#TableName, HelpText, Chronological, updatePeriod, updateEvent
RateLimit_ctl,NULL,n,5m,exc;

#FieldSchema/UserFieldSchema
#Fieldname, Label, FieldType, DataType, ChoiceName,RangeName,
   Keyfield,NotNull;
EMSName,,nearend,varchar2(16),,,y,n,EMS Identifier;
DeviceName,,ref,varchar2(32),,,y,n,Device Name;
Partition,,ref,varchar2(32),,,y,n,NexTone partition name;
Bucket,,ref,varchar2(32),,,y,n,NexTone queue name;
DeviceMode,,ref,varchar2(32),,,y,n,NexTone mode;
Layer,,ref,varchar2(32),,,y,n,NexTone layer;
LimitType,,ref,varchar2(16),,,y,n,Limit Type;
LimitRate,,raw,number,,,n,n,Steady-state rate limit;
LimitBurst,,raw,number,,,n,n,Burst rate limit;
UlogRate,,raw,number,,,n,n,Ulog rate limit;
Threshold,,raw,number,,range_nextone_limit,n,n,Threshold;
```

Figure 10 Example of the *“common”* file.

```
#JoinTables - optional entry for setting up joins
#DataTable, RefTable, DataCols, RefCols, JoinClause, JoinType;
"OCLENDPNT_CTL","RATELIMITPOLICY_CTL","EMSNAME","EMSNAME","OCLENDPNT_CTL
   .RATELIMITPOLICY=RATELIMITPOLICY_CTL.RATELIMITPOLICY and
   OCLENDPNT_CTL.SBCNAME=RATELIMITPOLICY_CTL.SBCNAME and
   OCLENDPNT_CTL.EMSNAME=RATELIMITPOLICY_CTL.EMSNAME",;
```

Join Files

References

“Creating a UDDMType” (p. 8) in the *System Administration Guide*

Related command(s)

“manage_ccrules” (p. 13)

“create” (p. 5)

“dbtest” (p. 17)

“manage_aggr” (p. 46)

□

manage_udneitype

Description

The manage_udneitype command allows you to load or reload the contents of the *"/musr/rb/udnei/udneitype"* file. This command validates input file and ensures that the UDNEIType directory exists under *"/musr/rb/udnei/<dcol>"*.

Syntax

```
manage_udneitype act UDNEIType
```

Parameters

act — This parameter defines a specific action. Valid values include:

- add - creating UDNEIType in the system,
- del - removing UDNEIType from the system,
- mod - recreating UDNEIType in the system (remove then create),
- test - testing status of the specific UDNEIType.

UDNEIType — Full UDNEIType name.

System responses

The following table shows the responses for the manage_udneitype command.

IF the command is ...	THEN the system responds with ...
executed, but other instance of manage_udneitype is running.	`manage_udneitype is already running.`
executed with *act=add* and given *<udneitype>* exists.	`You cannot perform add operation on existing udneitype.`

Result

After successful execution of the manage_udneitype command all mappings between UDNEI type given by <UDNEIType> and <dcols> are added, removed, modified in the Oracle Database. This command creates *"/musr/rb/udnei/<dcol>/<UDNEIType>"* and *"/musr/swdata/udnei/<UDNEIType>"* folders.

Examples

The following are the examples of a `manage_udneitype` command entry:

```
$ manage_udneitype add my_UDNEIType
$ manage_udneitype del my_UDNEIType
```

Template File

The following is the template for the *"/musr/rb/udnei/udneitype"* file. Figure 11 shows the example of that file.

```
udneitype=<udneitype>,dcol=<dcol>
```

Template Parameters

Below is a list of the parameters used in the *"/musr/rb/udnei/udneitype"* file:

`udneitype`	name of udneitype (up to 30 characters).
`dcol`	DCOL name (up to 30 characters).

Template Example

Figure 11 Example of the *"/musr/rb/udnei/udneitype"* file.

```
udneitype=mgip, dcol=simpleftp
udneitype=uaip, dcol=normalizer
udneitype=uaip, dcol=simpleftp
```

References

"Using an existing UDNEI Data Collector" (p. 11) in the *System Administration Guide*.

Related Command(s)

"manage_dcol_params" (p. 23)

"manage_ccrules" (p. 13)

□

udnei_act

Description

The `udnei_act` command allows you to start the data collectors operated on behalf of the specific UDNEIType.

Syntax

```
udnei_act UDNEIType
```

Parameters

`UDNEIType` Full UDNEIType name.

Examples

The following is an example of the `udnei_act` command.

```
$ udnei_act my_UDNEIType
```

References

"Provisioning a network element instance" (p. 16) in the *System Administration Guide*.

□

udnei_deact

Description

The `udnei_deact` command allows you to stop the data collectors operated on behalf of the specific UDNEIType.

Syntax

```
udnei_deact UDNEIType
```

Parameters

`UDNEIType`	Full UDNEIType name.

Examples

The following is an example of the `udnei_deact` command.

```
$ udnei_deact my_UDNEIType
```

References

Chapter 16, “UDDM/UDNEI Administration” in the *System Administration Guide*.

□

manage_aggr

Description

This command is only available if you have purchased Feature 460, “Periodic Data Aggregation”.

The `manage_aggr` command configures the periodic data aggregation engine.

Syntax

```
manage_aggr [option] action UDDMType1 [UDDMType2,...]
```

Parameters

`action` Parameter which defines specific action. You can put one of the following:

- act/deact – activate or deactivate aggregation.
- add – add aggregation for particular *`UDDMType`*.
- del – delete aggregation for selected *`UDDMType`*.
- edit – edit file for specified *`UDDMType`*. The text editor is determined by the `EDITOR` environment variable. If this variable is not set, `vim` is assumed.
- initfile – create initial file */musr/uddm/tables/<UDDMType>_aggr*. User is required to define aggregation levels and formulas in this file. For example of *<UDDMType>_aggr* file, see “<UDDMType>_aggr File Example” (p. 16-26)
- reschedule – reschedule already created aggregation jobs to different offset relative to aggregated period end.
- restart – restart aggregation jobs for already created aggregations which are not in “scheduled” state.
- status – displaying status of the aggregation.
- test – test of already loaded formulas for selected aggregation.

`option` This parameter defines a specific option. Valid values include:

- `-h, --help` – shows help message
- `-v, --verbose` – prints additional information to stdout.
- `-f, --force` – forces action.

`UDDMType` This parameter defines UDDM data type. You can use `all` instead of selecting particular UDDM data type.

Examples

The following examples present usage of the `manage_aggr` command.

```
$ manage_aggr status all
$ manage_aggr add all
$ manage_aggr del sbc
$ manage_aggr initfile sbc
```

Sample Output

```
$ manage_aggr status all
Model Type Act    Status     Last run     Next run TS Free  Size Days  Ret
----- ---- --- --------- ----------- ----------- ------- ----- ---- ----
  DTN    H   y scheduled 08/07 09:55 08/07 10:55    1 GB 16 KB   7d  45d
  DTN    D   y scheduled 08/07 01:01 09/07 01:01    1 GB   122   7d  90d
  DTN    M   y scheduled             01/08 04:00    1 GB     ?   0d 360d
```

References

Chapter 16, "UDDM/UDNEI Administration" in the *System Administration Guide*.

Related Command(s)

"manage_uddm" (p. 30)

□

manage_stat

Description

This command is only available if you have purchased Feature 461, “Statistical Thresholds”.

The `manage_stat` command configures the statistical thresholding engine. The statistical thesholds are derived from data saved by a separate Feature 460, “Periodic Data Aggregation”.

Syntax

```
manage_stat action UDDMType [option]
```

Parameters

`action` Parameter which defines specific action. You can put one of the following:

- get – this action will query and display the historical mean. The action requires the object identifier, and a 5-minute point in time.
- suspect – tagging bad data after it has been collected. The action requires date, time range and data type level. This action requires to pass additional parameter:
  - add – add suspected period.
  - del – delete a suspected period.
  - list – list suspected period.
- add – creating statistical thresholding objects in the system based on the input file */musr/uddm/tables/<UDDMType>_aggr.*
- del – removing statistical thresholding objects from the system.
- mod – recreating statistical thresholding objects in the system (remove then create).
- status – this action presents status of defines statistical thresholds. To present status of all defined statistical objects you can use `all` as a value of the *`UDDMType`* parameter.
- recompute – this action forces recalculation of the historical means for *`UDDMType`*.   Use this parameter when you either changed tags on suspected data or modified the list of the atypical days.

`UDDMType` This parameter defines UDDM data type.

`option` This parameter defines a specific option. Valid values include:

- `-h, --help` – shows help message
- `-q, --quiet` – prints only errors on stdout.

- `-k KEYVALUES, --keyvalues=KEYVALUES` – valid only for *`action`* suspect and get. List of keys=value pairs e.g:
  -k emsname=tstrsm51_4,sbcname=MSx2,sbcid=2
- `-p PERIOD, --period=PERIOD` – valid only for *`action`* suspect and get. Period range, e.g. –p 05/23/2011-06:00,05/23/2011-07:00

Examples

The following examples present usage of the `manage_stat` command.

- Create a statistical tresholding object for the sbc *`UDDMType`*

```
$ manage_stat add sbc
```

- Query and display the historical mean for the sbc *`UDDMType`* for a given period and the following key field

```
$ manage_stat get sbc -p 05/23/2011-05:00 -k
  emsname=tstrsm51_4,sbcname=MSx2,sbcid=2
```

- List suspected data for the sbc *`UDDMType`* and the following key fields for a given period

```
$ manage_stat suspect list sbc -k
  emsname=tstrsm51_4,sbcname=MSx2,sbcid=2 -p 05/23/2011-05:00
```

- Tag data as suspected for the sbc *`UDDMType`* and the following key fields for a period range

```
$ manage_stat suspect add sbc -k emsname=tstrsm51_4,sbcname=MSx2,sbcid=2
  -p 05/23/2011-06:00,05/23/2011-07:00
```

- Present the status of all statistical objects for *`UDDMType`*

```
$ manage_stat status all
```

- Force recalculation of the historical means for the SBC *`UDDMType`*

```
$ manage_stat recompute sbc
```

References

Chapter 16, "UDDM/UDNEI Administration" in the *System Administration Guide*.

Related Command(s)

"manage_uddm" (p. 30)

"manage_ccrules" (p. 13)

"manage_aggr" (p. 46)

□

9 Status and Other Commands

Overview

Purpose

This chapter lists and describes NTM status commands and those that have been categorized as “other”. Other commands include status commands (used to obtain the status of NTM or network elements), commands that allow you to send messages to other users, etc.

Contents

This chapter discusses the following topics:

□

Command functions

Table

Table 1 lists each NTM command categorized as status and “other” and provides a description of its function.

Table 1 Status and “other” commands and functions

Command	Function
`datastat`	Shows the data transmission status for NTM offices
`dsc`	Monitors discretes
`errlog`	Views error log files
`guiwall`	Allows users to send out broadcast messages from the host machine
`linkstat`	Displays data collection status of offices and DCC(s)
	Lists all the trunk groups that have calculations inhibited
`sendmsg`	Verifies end-to-end connection between NTM and a DCC
`snw_info`	Gives status information on subnetworks
`startsys`	Starts or restarts the system
`sysstat`	Reports information about the current state of the system

□

datastat

Description

The `datastat` (Office Data Status) command shows the data transmission status for NTM offices.

Syntax

```
datastat  office=office
```

Before you begin

Make sure the system is running prior to running the `datastat` command. If the system is not running, you will receive the appropriate error message.

Parameters

`office`	Office name(s)

System responses

```
$datastat

xchnge = simrss3

 Data for period Ending on Mon Apr  6 15:30:00 1998
Period = 891890700, Timeslot = 233

**
**OUTPUTTING DATA FOR ENTITY  axeprxemm (1:1:25)
**

RAWBLOCKS=221895, -1, -1, -1, -1, -1
PACKET NAME                      BLOCK OFFSET LENGTH FLAGS
Delayed Readiness                  0     992      4  GOOD
Overload                           0       0      9  GOOD
Call Direction                     0       9    178  GOOD
Matching Loss No Circuit           0       0      3  GOOD
Service                            0     111      3  GOOD
Ineff Mach Att                     0      30      6  GOOD
Control Counts                     0       0      3  GOOD
CCS/Machine Activity               0      24      5  GOOD
IEC Shared TG Data                 0      10     15  GOOD
Trunk Group                        0     187    228  GOOD
TG Flags                           0     415     35  GOOD
System Performance                 0     450    120  GOOD
Peripheral Unit Performance        0     570    120  GOOD
```

```
Host-Remote Link Perf              0     690     60  GOOD
Code Controls                      0     750      2  GOOD
Manual Reroute Controls            0     752    240  GOOD
TPC Counts                         0      24      5  GOOD
GETS Office Counts                 0      10     15  GOOD
GETS Trunk Group Counts            0     187    228  GOOD
           simrss3: Total Number of Bytes= 1992
---------------------------------------------------------
```

Examples

```
$  datastat  office=clevdohsn_04
```

Related command(s)

“linkstat” (p. 9)

□

dsc

Description

The dsc (monitor discretes) command allows you to monitor discretes from one or more specified offices on a continuous basis. It is normally used as a tool to look at the raw discrete data from network elements. Users view this information via the Network Elements, Discretes page. The number of discretes a network element can send to NTM varies according to the type of network element.

The basic output from the dsc command indicates the status of the collection of the discrete data, the actual hex layout of the discrete data, and a legend telling which bit numbers are "on" in the discrete data as well as there associated meaning. The meanings associated with each bit will vary according to the type of network element.

Syntax

```
dsc office=office name
```

Parameters

office name   A list of one or more offices to be monitored

Important! You can enter multiple values separated by the plus sign (+).

System responses

The output of the dsc command will continue every 30 seconds until you press **Delete**.

Figure 1 shows the responses for the dsc command.

Figure 1 dsc system responses

IF the command is …	THEN the system responds with …
executed successfully	IP followed by the output shown in Figure 2; this output shows the discrete status for each requested office in two forms: • a hexadecimal representation of the entire discrete bitmap for the office • a list of discrete mnemonic/description pairs for each discrete that is set, that is, for each discrete that is not at zero.
executed but discretes are manually out of service for this office	MAN_OOS
executed but the data collector is manually out of service	ED_MAN_OOS

Figure 2 shows the output for the `dsc` command. The output of the `dsc` command will continue every 30 seconds until you press **Delete**.

Figure 2 dsc command output

```
Discrete data collected for the period Mon Apr 10 14:59:00 2000
  alhbca0157h : DCSUCCESS    44   04   40   90   00   00   00   00
        02 DSM1DP      - MC1-DP thresh. exceeded
        06 DSM2DP      - MC2-DP thresh. exceeded
        10 TRMDOC      - SILC in effect, transmitting DOC signals
        22 RADRDP      - DP RADR delay
        28 TDPRCQ      - tk DP recvr queue ovfl
        31 CDPRCQ      - cust. DP rcvr queue ofl

Discrete data collected for the period Mon Apr 10 14:59:00 2000
  elmnca01cg0 : DCSUCCESS    00   00   00   00   00   40   00   00
        46 RMVDOC      - audit removed incoming overload control (SILC or DOC)
```

Examples

The following example shows the `dsc` command being run for two offices.

```
$ dsc office=austtxgr016t+hrlntxat02t
```

□

errlog

Description

The errlog (error log) command allows you to view error log files without knowing the error log file names.

Syntax

```
errlog [version=ver]
```

Parameters

ver Version of Error Log. Choose one:

- current (default)
- old

System responses

An IP acknowledgment is followed by an invocation of the more command using the error log file.

Examples

```
$ errlog version=old
```

References

See the *System Responses Guide* for more information on the error messages in the log.

□

guiwall

Description

The `guiwall` command allows users to send out broadcast messages from the host machine.

- When used with the `[-all]` option, the message is sent to all users on both the NTM host and GUI.
- When used without the `[-all]` option, the message is sent to users on the NTM GUI only.
- Use **CTRL-D** to end the input message and send it.

For reference, see *`Broadcast Messages`*.

Syntax

```
guiwall [-all]
```

System responses

Figure 3 shows the responses for the `guiwall` command.

Figure 3 guiwall system responses

IF the following command is executed successfully ...	THEN the system responds with ...
`guiwall`	the acknowledgment: The following message has been sent to GUI users: <*message*>
`guiwall -all`	an acknowledgment similar to the *Linux* wall command: <*Broadcast message from* ...>

Related command(s)

`wall` (*Linux* command)

□

linkstat

Description

The `linkstat` (link status) command displays the data collection status of office(s) and DCC(s).

Syntax

```
linkstat [office=office] [type=type] [parse=parse]
   [ofcset=ofcset] [dcc=dcc] [dconn=dconn]
```

Passwords / logins

The `linkstat` command cannot be run by the “`root`” user.

Parameters

`office` Office name(s), and/or DCC names, or all; “all” is the default.

Reference: Note 1

`type` The type of offices to display:

- all = status for all types is shown; “all” is the default
- dcc = status for all DCC offices
- ess1a = status for all 1A *ESS* offices
- ess4 = status for all *4ESS* offices
- ess5 = status for all *5ESS* offices
- ewsd = status for all *EWSD* offices
- dms = status for all *DMS* 100/200 offices
- dms250 = status for all *DMS* 250 offices
- dms300 = status for all *DMS* 300 offices
- dms500 = status for all *DMS* 500 offices
- gsx = status for all Sonus GSX offices
- gtd5 = status for all *GTD-5* offices
- lssgr = status for all LSSGR offices
- psx = status for all Sonus PSX offices
- scsnsn = status for all SCSNSN offices
- nms = status for all NTM offices
- udne = status for all UDNEI network elements

Reference: Note 1

`parse` A value of “Y” means never leave a column blank, so the GUI can count on a specific number of columns per line.

This parameter is of little use from the command line; it was added for internal GUI use.

Valid values are: Y, YES, N, NO

`ofcset` List of office sets. This parameter is provided so the GUI can supply a SET field, and the user can type in a set and get a `linkstat` on all offices in one or more of those sets.

Reference: Note 1

`dcc` List of DCCs. This parameter is provided so the GUI can allow the user to type in a DCC or Mediation Device name and get status on all the switches attached via that device plus the device itself.

Reference: Note 1

`dconn` A value of “Y” produces a list of all directly connected switches.

Valid values are: Y, YES, N, NO.

Notes

1. You can enter multiple values separated by the plus sign (+).

Examples

This example displays the status for switch “abcd”:

```
$ linkstat office=abcd
```

This example displays the status for all *4ESS* switches:

```
$ linkstat type=ess4
```

System responses

The system responds with an IP acknowledgment, followed by the status of each office. If NTM is not running, a message indicating this will be displayed. Table 2 lists the system response examples available for this command.

Table 2 linkstat figures

Figure	type=
Figure 4	`ess4`
Figure 5	`ess1, ess1a, ess5, dms, dms250, dms300, dms500, ewsd, gtd5, or lssgr`
Figure 6	`dcc`

Table 2 linkstat figures

Figure	type=
Figure 7	nms
Figure 8	dconn, scsnsn

4ESS switch link status

If type=ess4, the output appears as shown in Figure 4.

Figure 4 linkstat command output — 4ESS — systems with BDR

```
$ linkstat type=ess4
  IP

    4ESS             ACTIVATION STATUS    CONNECTION STATUS     INTER-NMS
    ----             ----------------- -----------------------  ----------
    CLLI     TYPE     AUD MEA DSC CTL        SESSION SYNC TIME  TKO PRIMARY

okldca0349t  4e12      -   -   -   -         idle    -     -     -     -
tst4e23      gen4e    act act act act        conn    -    -0002  -     -
sntcca0148t  4e12     act act act act        conn    sync -0002  -     -
anhmca0295t  4e15     act act act act        conn    -    -0002  -     -
shokca0589t  4e14     act act act act        conn    -    -0002  -     -

sndgca0291t  4e18     act act act act        conn    -    -0002  -     -
$
```

In Figure 4, the column headings have the following meanings:

Column Heading	Subheading
CLLI	CLLI code name
TYPE	Office type and generic
ACTIVATION STATUS	Either act (activated) or "-" (deactivated) is shown under the following subheadings: • AUD = Audits • MEAS = 5-minute data • DSC = Discretes • CTL = Controls
(Sheet 1 of 2)	

Column Heading	Subheading	
CONNECTION STATUS	Following are the subheadings under CONNECTION STATUS along with the entries that may be found in each respective column:	
	SESSION	• idle = connection is down, normal • conn = connected • conn-p = connection pending • failed = connection is down, abnormal • disc-p = disconnect pending • invalid = invalid
	SYNC	• sync = NTM is automatically synchronizing its time to the time of this office • inh = automatic synchronization is inhibited (i.e. "/nm/db/notimesync" file exists) • - = the office is not the sync office
	TIME (Time difference in seconds between the office and NTM)	• +*n* = indicates DCC is ahead of NTM by "n" seconds up to a maximum of +30 seconds allowable • -*n* = indicates DCC is behind NTM by "n" seconds up to a maximum of -15 seconds allowable • - = indicates office time is not available
	⚠ **WARNING** **Failure to keep the DCC within the maximum ± seconds allowable can result in data collection failure for offices reported by this DCC.** **Important!** Time difference indications do not include time zone differences.	
INTER-NMS (Inter-NMS communications)	Following are the subheadings listed under INTER-NMS along with the entries that may be found in each respective column:	
	TKO (Takeover mode)	• - = Your host is not in takeover mode • act = takeover mode has been activated
	PRIMARY (The host considered primary for that switch)	• - = The NTM host executing the command is primary • `other_host` = Another host is considered primary for that switch • none = No primary has been defined for that switch
(Sheet 2 of 2)		

Switch by way of DCC link status

If type=ess1, ess1a, ess5, dms, dms250, dms300, dms500, ewsd, gtd5, or lssgr, the output appears as shown in Figure 5.

Figure 5 linkstat command output — BDR connected through DCC

```
 DCC  OFC         ACTIVATION STATUS  SWITCH STATUS AT DCC      INTER-NMS
---------         ----------------- ------------------------- ---------
    CLLI      TYPE   AUD MEAS DSC CTL   DCC         MEAS   DSC   TKO PRIMARY

bsrnca70ds0  sn04   act act  act act fep_2n       online  good  -     -
anhmca01ds0  sn06    -   -    -   -    -           -       -    -   cbnmvd
anhmca11ds0  dms26    -   -    -   -    -           -       -    -   cbnmvd
argrca12ds0  dms28    -   -    -   -  dcc1n       -       -    -     -
bkfdca1176t  dms28   act act  act act dcc4n      online  good  -     -

bkfdca19ds0  5e6      -   -    -   -  fep_2n        -       -    -     -
bklyca01ds0  gtd1711  -   -    -   -    -           -       -    -     -
```

In Figure 5, the column headings have the following meanings:

Column Heading	Subheading
CLLI	CLLI code name
TYPE	Office type and generic
ACTIVATION STATUS	Either act (activated) or "-" (deactivated) is shown under the following subheadings: • AUD = Audits • MEAS = 5-minute data • DSC = Discretes • CTL = Controls

Column Heading	Subheading	
SWITCH STATUS AT DCC	The subheadings listed under this heading along with the entries that may be found in each respective column are as follows:	
	DCC (The DCC to which the office is connected)	To fully activate data collection, you must activate a data type for both the office and the corresponding DCC. Look at the DCC link status to determine the connection status.
	MEAS (5-minute data)	• online = Office is on-line at DCC • offline = Office is manually off-line at DCC • timeout = DCC cannot collect data • suspect = DCC has marked 5-minute data as being suspect • invalid = DCC does not know about the office
	DSC (The discrete status of the switch at DCC)	• good = DCC successfully collecting discretes • fail = DCC is not able to collect new discretes
INTER-NMS	(Inter-NMS communications) Following are the subheadings listed under INTER-NMS along with the entries that may be found in each respective column:	
	TKO (Takeover mode)	• - = Your host is not in takeover mode • act = takeover mode has been activated
	PRIMARY (The host considered primary for that switch)	• - = The NTM host executing the command is primary • *other_host* = Another host is considered primary for that switch • none = No primary has been defined for that switch

DCC link status

If `type=dcc` the output appears as in Figure 6.

Figure 6 linkstat command output for DCCs

```
DCC              ACTIVATION STATUS       CONNECTION STATUS       INTER-NMS
-----             -----------------  --------------------------- ------------
   CLLI      TYPE   N AUD MEAS DSC CTL AUD   MEAS  DSC   CTL  TIME  TKO PRIMARY

dcc0u        1bed6  D  -   -   -    -  idle  idle  idle  idle  -     -     -
dcc1u        1bed6  D act act act act fail  fail  fail  fail  -     -     -
dcc3u        1bed6  D  -   -   -    -  idle  idle  idle  idle  -     -     -
dvdcp1       tdms2  D act act act act conn  conn  conn  conn  -     -     -
fep_tcp      fepr1  T act act act act conn  conn  conn  conn +0001 -     -

db1bed6ent31 1bed6  D  -   -   -    -  idle  idle  idle  idle  -     -   cbnmsb
```

```
dccxr        1bed6  D  -   -   -    -  idle  idle  idle  idle  -     -  cbnmsb
dcc0n        1bed6  D  -   -   -    -  idle  idle  idle  idle  -     -  cbnmsb
dcc1n        1bed6  D  -   -   -    -  idle  idle  idle  idle  -     -  cbnmsb
dcc2n        1bed6  D  -   -   -    -  idle  idle  idle  idle  -     -  cbnmsb

fep2         fepr1  D  -   -   -    -  idle  idle  idle  idle  -     -  cbnmsb
fep1a        fepr1  D  -   -   -    -  idle  idle  idle  idle  -     -  cbnmsb
dcp1         fepr1  D  -   -   -    -  idle  idle  idle  idle  -     -  cbnmsb
stfep1a      fepr1  D  -   -   -    -  idle  idle  idle  idle  -     -  cbnmsb
cbtdms5      fepr1  D  -   -   -    -  idle  idle  idle  idle  -     -  cbnmsb

dcc1s        1bed6  D  -   -   -    -  idle  idle  idle  idle  -     -  cbnmvd
dcc3s        1bed6  D  -   -   -    -  idle  idle  idle  idle  -     -  cbnmvd
dcc4s        1aed8  ?  -   -   -    -  idle  idle  idle  idle  -     -  cbnmvd
dcc5s        1aed8  ?  -   -   -    -  idle  idle  idle  idle  -     -  cbnmvd
dcc6s        1aed8  ?  -   -   -    -  idle  idle  idle  idle  -     -  cbnmvd
```

In Figure 6, the column headings have the following meanings:

Column Heading	**Subheading**
CLLI	CLLI code name
TYPE	DCC type and generic
ACTIVATION STATUS	Under the first subheading (N), a value of either D, T, or (?) will appear. Either act (activated) or "-" (deactivated) is shown under the following subheadings: • AUD = Audits • MEAS = 5-minute data • DSC = Discretes • CTL = Controls

Column Heading	**Subheading**	
CONNECTION STATUS	Link status for each DCC.	
	AUD **MEAS** **DSC** **CTL**	For each of these subheadings, the entries that may be found in each respective column are: • idle = connection is down, normal • conn = connected • conn-p = connection pending • failed = connection is down, abnormal • disc-p = disconnect pending • invld = invalid
	TIME (Time difference in seconds between DCC and NTM)	• +*n* = indicates DCC is ahead of NTM by “n” seconds up to a maximum of +30 seconds allowable • -*n* = indicates DCC is behind NTM by “n” seconds up to a maximum of -15 seconds allowable • - = indicates DCC time is not available
	WARNING **Failure to keep the DCC within the maximum ± seconds allowable can result in data collection failure for offices reported by this DCC.** **Important!** Time difference indications do not include time zone differences.	
INTER-NMS (Inter-NMS communications)	The subheadings under INTER-NMS and the possible entries for each respective column are:	
	TKO (Takeover mode)	• - = Your host is not in takeover mode • act = takeover mode has been activated
	PRIMARY (The host considered primary for that switch)	• - = The NTM host executing the command is primary • *other_host* = Another host is considered primary for that switch • none = No primary has been defined for that switch

NMS office link status

If `type=nms`, the output appears as in Figure 7.

Figure 7 linkstat command output for NMS hosts

```
NMS            ACTIVATION STATUS    CONNECTION STATUS
```

```
----           -----------------   ---------------------
CLLI             LOCAL      REMOTE   SESSION      TAKEOVER

cbnmsc             act         ?        fail             -
btsun              act         ?        fail             -
```

In Figure 7, the column headings have the following meanings:

Column Heading	**Subheading**	
CLLI	CLLI code name	
TYPE	DCC type and generic	
ACTIVATION STATUS	Following are the subheadings along with the entries that may be found in each respective column:	
	LOCAL (Status of link to the other NMS center)	• act = activated • - = not activated
	REMOTE (Status of link at both ends)	• act = link is up and active on both ends • ? = link is down or manually deactivated
CONNECTION STATUS	Link status for each NMS. Following are the subheadings along with the entries that may be found in each respective column:	
	SESSION	• idle = link is manually deactivated • conn = both NTM hosts are up and communication is established • fail = Link is activated, but communication has not been established
	TAKEOVER	• act = host is in takeover mode • - = host is not takeover mode

Non-4ESS direct-connect office link status

If `type=dconn` (and generics na013, ncs13 and ucs13 and later are available) or `type=scsnsn`, the output appears as in Figure 8.

Figure 8 linkstat command output for direct connected hosts

```
$ linkstat type=scsnsn
```

```
  IP

 SCSNSN OFC           ACTIVATION STATUS        CONNECTION STATUS    INTER-NMS
 ----------           -----------------------  ------------------   -----------
   CLLI      TYPE     AUD MEA DSC CTL SEC CSV  HI   MED  LOW  TIME  TKO PRIMARY

tstscsnsn    sn06     act act act act  -   -  idle idle idle  -      -     -
$
```

In Figure 8, the column headings have the following meanings:

<table>
<tr><th>Column Heading</th><th colspan="2">Subheading</th></tr>
<tr><td><strong>CLLI</strong></td><td colspan="2">CLLI code name</td></tr>
<tr><td><strong>TYPE</strong></td><td colspan="2">Office type and generic The type field is limited to 4 display characters, although more may be defined.</td></tr>
<tr><td><strong>ACTIVATION STATUS</strong></td><td colspan="2">Either act (activated) or “-” (deactivated) is shown under the following subheadings:
• AUD = Audits
• MEAS = 5-minute data
• DSC = Discretes
• CTL = Controls
• SEC = Security
• CSV = Comma Separated Value (SCSNSN sn06+)</td></tr>
<tr><td rowspan="5"><strong>CONNECTION STATUS</strong></td><td colspan="2">The subheadings listed under this heading along with the entries that may be found in each respective column are as follows:</td></tr>
<tr><td><strong>HI</strong>
(Discretes and controls)</td><td rowspan="3">• idle = link is manually deactivated
• conn = the channel between NTM and the switch is up and communications have been established
• fail = Link is activated, but communication has not been established.
• <strong>NOTE</strong>: The MED status will show failed if the TCP or FTP status is not established on SCSNSN sn06+ offices.</td></tr>
<tr><td><strong>MED</strong>
(5-minute data)</td></tr>
<tr><td><strong>LOW</strong>
(Audits)</td></tr>
<tr><td><strong>TIME</strong>
(Time difference in seconds between office and NTM)</td><td>• +<em>n</em> = indicates office is ahead of NTM by “n” seconds up to a maximum of +30 seconds allowable
• -<em>n</em> = indicates office is behind NTM by “n” seconds up to a maximum of -15 seconds allowable
• - = indicates switch time is not available</td></tr>
</table>

Column Heading	Subheading	
INTER-NMS (Inter-NMS communications)	Following are the subheadings listed under INTER-NMS along with the entries that may be found in each respective column:	
	TKO (Takeover mode)	• - = Your host is not in takeover mode • act = takeover mode has been activated
	PRIMARY (The host considered primary for that switch)	• - = The NTM host executing the command is primary • *other_host* = Another host is considered primary for that switch • none = No primary has been defined for that switch

□

sendmsg

Description

The sendmsg (send message) command allows you to transmit a text message to a DCC to verify an end-to-end connection between NTM and a DCC. The message is sent from NTM to DCC over the “control” connection.

Verification of the connection requires working with someone at the DCC. The DCC displays text messages received by NTM on its system console. If the DCC displays the message that was sent from NTM, the physical connection between DCC and NTM is verified as “working.”

Syntax

```
sendmsg entity=dcc
```

Parameters

dcc	Name of the DCC to which you are sending the message

Important! You can enter multiple DCC names separated by the plus sign (+).

System responses

Figure 9 shows the responses for the sendmsg command.

Figure 9 sendmsg system responses

IF the command is ...	THEN the system responds with ...
executed successfully	IP and prompts for the text message

Figure 9 sendmsg system responses

IF the command is ...	THEN the system responds with ...
executed, but NTM is unable to send the message	one of the following messages: ?E INVALID OFFICE: dcc1 does not exist. Reenter value(s) for this keyword. entity = To correct the error, try the following: 1. Enter a question mark (?) to get a listing of valid DCC names and enter a valid name OR NG control connection for `dcc` is manually taken out of service 2. Activate the control connection for the specified DCC. OR NG control connection for `dcc` is failed taken out of service 3. Examine the system error log for possible problems with DCC. OR NG control connection for `dcc` is busy. Try again later. 4. In this message, a user is attempting to send controls through `dcc`. Use `sendmsg` command later.
is executed with the option "this is a test" after "message ="	IP message = this is a test

References

See the `act` command (7-4) for more information about activating the control connection.

For more information about performing a link diagnostic test program, see Chapter 10, "Time Synchronization" in the *System Administration Guide*.

Related command(s)

"act" (p. 4)
"deact" (p. 11)
"errlog" (p. 7)

□

snw_info

Description

The snw_info (subnetwork information) command can be used to display information on subnetworks and user groups. When you enter the snw_info command, you are provided with a display of current subnetworks, user groups, and user group permissions.

Syntax

```
snw_info
```

System responses

Figure 10 shows the output of the snw_info command.

Figure 10 snw_info Command Output

```
$ snw_info
  IP

       ****************************
       **                        **
       ** SUBNETWORK INFORMATION **
       **                        **
       ****************************

SUBNETWORKS:
pbno(M) uswe(P) pbxr(P) pbso(P) gtd5(P)

GROUP     HOME   PERMISSIONS

nm        pbno  pbno
                pbno&
rb        pbno  pbno
                pbno&
usr       pbno  pbno

*  surveillance-only permission.
&  database modification permission.
```

> **Important!** The line, `&    database modification permission`, is displayed only when Feature 3, “Management of Record Base Partitions and Subnetworks” is on.

References

Chapter 11, “Subnetwork Administration” in the *System Administration Guide*

Related command(s)

"snw_admin" (p. 17)

□

sysstat

Description

The sysstat (system status) command reports the following:

- The time the machine was last booted
- Whether NTM is running or stopped, and when it started or stopped
- The current database name
- When the last create all was run
- When the last build db was run
- When the last installdb all or recreate was run

Syntax

```
sysstat
```

System responses

Figure 11 shows the responses for the stopsys command.

Figure 11 sysstat Command Output

```
$ sysstat
  IP

The machine was last booted: Thu Jun 30 10:55:56 2005

The system is RUNNING
        started at: Tue Jul  5 02:35:57 2005

'CREATE ALL               ' last run: Thu Jun 30 11:36:19 2005
'BUILD DB                 ' last run: Thu Jun 30 10:59:45 2005
'INSTALLDB ALL or RECREATE' last run: Tue Jul  5 02:41:25 2005

The Oracle RDBMS is running
The 8920 NTM Relational DB software is running
$
```

See the create command (5-5) for more information on running a full create.

See the installdb command (5-34) for more information on running a full installdb.

□

Glossary

%	A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V	W

%%OCC Percent Occupancy

The fraction of time that a circuit or a piece of equipment is in use, expressed as a decimal. Numerically, it is the Erlangs carried, and it equals the carried CCS divided by 36. Percent occupancy measurements include both message time and setup time.

%OFL Percent Overflow

The relationship between the total attempts offered in a specific time period to a route or a destination and the number of attempts not finding an idle circuit.

AAB A-B trunk group

A trunk group that connects an originating office (A) directly to a terminating office (B). See "AV" (p. 3) and "VB" (p. 25).

ACC Automatic Congestion Control

Senses machine congestion and activates preplanned internal and external overload controls. Also called/see also DOC. See the `acc` command (4-9) in the *Input Commands Guide*.

ACG

Automatic Call Gap

ACH Attempts per Circuit per Hour

Relationship between the number of attempts that result in an answer signal and the total number of attempts.

ACM Address Complete Message

A messages sent in the backward direction indicating that all the address signals required for routing the call to the called party have been received.

Activate

To make an office active for data collection.

ADL-V

AT&T Digital Link — Phase 5

Aggregated Trunk Group

An aggregated trunk group is not a physical trunk group but rather a collection of all traffic information on trunk groups to a particular "to office", represented with a unique trunk group ID. In this way, controls can be sent to a 7R/E switch for a given "to office" by specifying the tg ID of the aggregated trunk group.

Aggregation Limit

Date and time limit you can set on the aggregation view to limit the number of records that will appear in your report.

AIC Available Idle Circuits

A traffic measurement used by network managers to determine which trunk groups have capacity available for rerouting traffic from an overloaded trunk group.

AIN Advanced Intelligent Network   Also called an Intelligent Network) A network:

- That affects the routing of calls within it from moment to moment based on a criteria other than simply finding a path through the network for the call
- Where the originator or the ultimate receiver of the call can inject intelligence into the network and affect the flow of his call (either outbound or inbound).

Intelligent networks generally include SCP, SSP, and STP components.

Alarm

Visible report of a trouble condition in the network. Alarms usually require immediate attention from network personnel.

Alert

Visible report of a potential trouble condition in the network.

Alerting Discrete

An on/off indicator that notifies network managers of changes to the status of the office. An alerting discrete provides a message to NTM that starts a corresponding audit (unless that audit has been previously inhibited by the network manager).

Allow

Indicates the permitting of an action, such as permitting automatically triggered audits to run.

Alternate Routed Traffic

Traffic that has been offered to a previous trunk group and has not been able to find an idle circuit. The switching system handling the traffic then offers it to an “Alternate Route,” based on its internal routing tables.

Alternate Routing

A means of selectively distributing traffic over a number of routes, ultimately leading to the same destination.

APC

Adjacent Point Code

APR Allow Previously Rerouted

A trunk group reroute control option that allows previously rerouted traffic to reroute. Only *4ESS* and *5ESS* offices support this reroute control option.

APS

Attached Processor System

ASCII American Standard Code for Information Interchange

A 7-bit code for providing as many as 128 different characters. An eighth bit can be added as a parity check for error detection purposes.

ASP

Advanced Services Platform

ATM Asynchronous Transfer Mode

A high bandwidth, low-delay, connection-oriented, packet-like switching and multiplexing technique that allows very high speed transmission.

Attempt

An attempt to seize a circuit in a route. An attempt may be successful or unsuccessful.

Audit

An integrity check through which NTM corrects differences between its own database and office databases.

AV

A-V (via) trunk groups. A trunk group that connects an originating office (A) to a via office (V). See “AB” (p. 1) and “VB” (p. 25).

BBacking Up

The process of copying data onto a separate medium for the purpose of data retention.

BDR Backup and Disaster Recovery

See Feature 8, “Disaster Recovery (Duplex)” and Feature 40, “Enhanced Disaster Recovery” in the *System Overview*.

Blocking

The inability of the calling party to be connected to the called party because either all suitable trunk paths are busy or a path between a given inlet and any suitable free outlet of the switching network is unavailable.

Broadcast Message

A text message sent out by personnel using the NTM to other users on the system.

CCalculation

Calculated counts used to signify changing network conditions and, when thresholded, to alert network managers to events that might require action to prevent excessive network congestion.

CAMA Centralized Automatic Message Accounting

Specific version of AMA in which the ticketing of toll calls is done automatically at a central location for several central offices.

CANF Cancel From

A post-hunt protective trunk group control that prevents a percentage of overflow traffic for a selected originating trunk group from advancing to any alternate route. See the `canf/cant/skip` command (4-13) in the *Input Commands Guide*.

CANT Cancel To

A pre-hunt protective trunk group control that prevents a percentage of traffic from accessing a selected destination trunk group. See the `canf/cant/skip` command (4-13) in the *Input Commands Guide*.

CCIS Common Channel Interoffice Signaling

Carries telephone signaling information along a path different from the path used to carry voice.

CCITT

Consultative Committee on International Telegraphy and Telephony

CCS Centi (Hundred) Call Seconds

A unit of traffic used to express the average number of calls or the average number of devices in use. One CCS is equal to the continuous load for 100 seconds. The CCS for an hour is 36.

CCS Common Channel Signaling

A form of signaling in which a group of circuits share a signaling channel.

CCS7-NA

North American Version of CCITT#7

CG Call Gap

A protective control that allows a fixed number of calls to succeed to a code (telephone number) in a 5-minute interval. See the `cg` command (4-21) in the *Input Commands Guide*.

CGX

Call Gaps with an IC prefix (*1AESS* only)

CICR Cancel In-Chain Return

A reroute trunk group control option. When set to YES, does not allow traffic to return to in-chain routing. When set to NO, allows traffic to return to in-chain routing.

CLI

Caller Line Identification

Client

A client uses the resources of another device (computer) or application. Client is another term for a PC on a local area network.

CLLI

Common Language Location Identifier

CNI

Common Network Interface

Code

A numbering system for telephone addresses, for example, 614-555-1234 (NPA-NXX-XXX).

Connection

An attempt for a circuit that succeeds in obtaining a circuit. Also called a seizure.

Container Page

One of the five basic types of pages used in the GUI. It displays the results of a search or a map of a network area.

Control Data
Data that describes the actual controls in place for the network.

CPE
Customer Premises Equipment

CPU
Central Processing Unit

CR
Critical Alarm

CR Circuit Reservation
An automatic trunk group control that reserves the last few trunks of a trunk group for critical users exclusively and eliminates the need to queue critical users for inter-switch trunks. See also/also called STR. See the `cr` command (4-32) in the *Input Commands Guide*.

Crash Dump
The output from the hardware registers, the hardware stack, and the CPU.

CRO Cancel Rerouted Overflow
A reroute trunk group control option that prevents overflow traffic on a via route (VB) from overflowing back to the direct route (AV). Not activating the CRO can result in an external loop.

CSL
Communications Software Launcher

Customer Premises Equipment
All telecommunications terminal equipment located on the customer premises.

DDatabase
A collection of data organized for rapid search and retrieval by a computer.

DCC
Data Collection Concentrator

DCE
Distributed Computing Environment

DCS
Display Construction Set

Deactivate
To make an office inactive for data collection.

Demand Data

Data retrieved by the `demand` command (5-21) from the system database. The User Report Writer feature and SQL files use this data to create informational reports.

Destination

A specified area or country in which the called subscriber is located. A destination is identified by its destination code (the digits used for routing the call).

Detail Page

One of the five basic types of pages used in the GUI. It provides information (such as reference data) on specific network elements or network connections.

Direct Routed Traffic

Traffic that is being offered to the trunk group for the first time, not having been previously offered to a different trunk group. This traffic, which has not alternate routed, is sometimes called “First Routed” traffic.

Discrete

An on/off indicator that notifies network managers that:

- Changes have been made to the status of the office
- Significant events have taken place within the office

NTM polls the offices for discretes at regular intervals.

Disk Array

A disk subsystem combined with management software that controls the operation of the physical disks and presents them as one or more virtual disks to the host computer.

DOC Dynamic Overload Control

Also called/see also ACC

Domain

A type of calling service, such as POTS (Plain Old Telephone Service), ACNT (*Accunet*), SDN (Software Defined Network), or ISDN (Integrated Services Digital Network).

Dot Profile (.profile)

A file located in your home directory that alters your default *Linux* system environment. You can use your .profile to define environmental variables such as your terminal type, prompt string, or mailbox address.

DP

Dial Pulse

DPT

Dynamic Packet Trunks

DPTPRI
Dynamic Packet Trunks Prioritization

DPTRES
Dynamic Packet Trunks Reservation

DPTTID
Dynamic Packet Trunks Terminal Identifier

DSC
Dynamic Service Control

DSDC Direct Services Dialing Capability
Network services provided by local switches interacting with remote databases via CCIS.

DTMF
Dial Tone Multifrequency

DTS
Dial Tone Speed

EEA Equal Access
A trunk group reroute option for switches that limits the reroute to equal access traffic.

EADAS Engineering and Administration Data Acquisition System
A system in which traffic data are measured at switching systems by electronic devices, transmitted to a centrally located minicomputer, and recorded on magnetic tape in a format that is suitable for computer processing and analysis. Performs data collection in NTM for certain switch types.

Erlang
A measurement of traffic load equal to the continuous occupancy of one circuit (or unit of equipment) for one hour. An Erlang can express the capacity of a system; for example, a trunk group of 30 trunks, which in a theoretical peak sense might carry 30 Erlangs of traffic, would have a typical capacity of perhaps 25 Erlangs averaged over an hour.

Error Code
An identification field used to identify the module or feature reporting the error. See the **ERR_CODE** field help file.

Error Log
The error log is a file that contains the error messages being generated by NTM. See the `errlog` command (9-7) in the *Input Commands Guide*.

Error Messages

System responses resulting from software-detected errors, changes in the system status, or non-executable commands.

Error Number

Number associated with error codes that help identify specific messages. See the **ERR_NUM** field help file.

ESP

Essential Service Protection Triggered

ESS

Electronic Switching System

ETR Easy To Reach

A code (telephone number) is determined to be easy to reach because the attempts and failures to the code do not exceed user-defined thresholds.

Exception

A calculation based on office or trunk group data that exceeds a user-defined threshold. It indicates an abnormal working condition in the network.

Exception Level

A number associated with an exception, indicating the severity or priority of the exception. High-numbered exception levels are more severe.

Exception Processing

Process used to collect raw data from the switch, perform calculations on the data, and, as a result, find exceptions based on predefined thresholds.

Exception Report

Formatted report of all exceptions that have occurred during the most recent 5-minute period.

Execution Error

The NTM GUI presents error messages in response to conditions such as improper permission, execution errors, etc. Execution errors are related to the execution of requests that affect the network elements to which the NTM host is connected (e.g., control requests or HTR administration).

External Network Element

A network element that is defined in the NTM Record Base but for which surveillance data is not received by NTM.

FFEP Front-End Processor

An application that acts as a DCC. Available with purchase of Feature 214, “FEP Release 4” or Feature 257, “FEP Release 5”.

FHC

Final Handling Code

Final Trunk Groups

A trunk group that acts as a final route for traffic. Traffic can overflow to a final group from high-usage groups that are busy. Traffic cannot overflow from a final trunk group. Calls that overflow a Final Trunk Group are terminated unless they are rerouted by an NTM Reroute control. See the `rr` command (4-44) in the *Input Commands Guide*.

FML Field Manipulation Language

A set of C-language functions for defining and manipulating data storage structures called fielded buffers.

FOO

A foo is a term universally substituted for something real when discussing ideas or presenting examples.

From Office

Internal network element that originates the trunk group.

FSD

Feature Specification Document

Full Create

The process of constructing the database itself (once the database files have been prepared) or making major database modifications through the use of the `create` command with no arguments. This process also modifies the offline database.

Full Trunk Group

A trunk group that does not overflow calls to another trunk group because enough trunks are provided to give an acceptable blocking probability.

GGeneric

The version released to provide specific services, features, or functions.

GETS

Government Emergency Telecommunications Service

GSC

Group Signaling Congestion

GSM

Global Switching Module

GUI Form Elements

The elements that appear within a form on a web page. Form elements may consist of a label and one or more fields when they are used outside a table. See "GUI form elements" (p. 20) in the *User Guide*.

Hhecto

A unit of measure meaning 10 to the power of 2.

High-Usage Trunk Group (HU)

A trunk group that is the primary direct route between two switching systems. The group is designed for high average occupancy. To provide an overall acceptable probability of blocking, an alternate route must be provided for overflow traffic.

Host Computer

Computer (machine) used to run the NTM.

HPC High Probability of Completion

A phase of GETS that extends the enhanced routing and priority service to LEC networks traversed by the call.

HT Holding Time

The average duration of phone calls.

HTR Hard-To-Reach

A code (telephone number) is designated as hard-to-reach because the number of attempts and failures to the code exceed user-defined thresholds. See Chapter 7, "Hard-To-Reach (HTR)" in the *System Overview*.

HU High Usage

A trunk group that is the primary direct route between two switching systems. The group is designed for high average occupancy. For an overall acceptable probability of blocking, an alternate route must be provided for overflow traffic.

Hunt Types

The three hunt types for reroutes are ***regular***, ***order***, and ***spray***.

- The regular hunt uses only one out-of-chain engineering route for the reroute. Order and spray hunts can have from two to seven out-of-chain engineering reroutes.

- For the order hunt, an ordinary route-advance pattern is specified for the out-of-chain engineering reroutes, and the same route is always used as the starting point for the trunk hunt.
- For the spray hunt, rerouted traffic is divided evenly among the out-of-chain engineering routes through a rotation scheme.

See the HUNT field help file.

Hysteresis

The minimum amount of change required to make a difference.

IICCH Incoming Connections per Circuit per Hour

The incoming peg count divided by the number of equivalent 2-way circuits.

IEC

InterExchange Carrier

IMA

Ineffective Machine Attempts

Immediate Reroute

A reroute that diverts calls to one or more specified via trunk groups prior to the hunting of the “reroute from” trunk group.

IMS

IP (Internet Protocol) Multimedia Subsystem

INA

Ineffective Network Attempts

Incoming Calls

Incoming trunk seizures at the office.

Inhibit

Indicates the blocking of an action, such as blocking automatically triggered audits from running.

Input Command

User-invoked instructions to a system, entered in the command shell. Also called an input message and command. See the *Input Commands Guide*.

Internal Calls

Originating calls intended to complete on lines served by the switch.

Internal Error Message
An error message reported in the error log and on the system console.

Internal Network Element
Network elements from which surveillance data is collected.

INWATS Inward Wide Area Telephone Service
A service that allows subscribers to receive calls from specified areas with no charge to the person who's calling.

IP
In Progress

IRR Immediate Reroute
A pre-hunt trunk group control option that causes a percentage of a specified type of traffic to be rerouted before it is offered to the regular in-chain trunk group.

ISA
Integrated Service Assurance

ISDN Integrated Service Digital Network
A set of standards for digital transmission over ordinary telephone copper wire as well as over other media. ISDN integrates analog or voice data together with digital data over the same network.

Issue
Office generic issue number.

ISUP Integrated Service Digital Network User Part
Defines the protocol and procedures used to set up, manage, and release trunk circuits that carry voice and data calls over the public switched telephone network (PSTN). ISUP is used for both ISDN and non-ISDN calls. Calls that originate and terminate at the same switch do not use ISUP signaling.

IWBM
Inter-working Bridge Measurements.

LLATA
Local Access and Transport Area

Launch page
One of the five basic types of pages used in the GUI. It is used to select high-level data types to monitor.

LEC
Local Exchange Carrier

Link Status
The signaling system connection status of an office.

LNP
Local Number Portability

Logical Database
A logical database consists of a computer program system database and a *Linux* operating system file area.

LRN
Location Routing Number

LSSGR
LATA Switching System Generic Requirements

MMB Maintenance Busy
Conditioning a circuit, a terminal, or a termination to be unavailable for service. When unavailable, it is generally necessary that it appear busy to circuits that seek to connect to it. Sometimes referred to as “make busy”. See the MB field help file.

MC
Machine Congestion Level

Menu Mouse Button
Mouse button used to display context-sensitive menus. (Usually the right mouse button.) Click the menu mouse button once to display the menu, then use the Select Mouse Button to select an item (or subitem) from the menu.

MF
Multifrequency

Mnemonic
Executable name used to access menus, menu items, and pages on the terminal screen. A mnemonic is a word or string that is intended to be easier to remember than the thing it stands for.

Monitoring
Comparing the traffic on selected trunk groups with assigned thresholds.

MSU
Message Signaling Unit

MTP Message Transfer Part

The part of the SS7 protocol that provides for basic routing of signaling messages between signaling points.

NNC

No Circuits

NCP Network Control Point

A routing, billing, and call control database system.

NEA Non-Equal Access

A trunk group reroute control option for switches that limits the reroute to non-equal access traffic.

Network Traffic Management

A system that provides near-real time surveillance of the network elements connected to it for the purpose of managing network congestion.

Network Data

Traffic data that is collected from the network elements on a periodic basis, typically 5 or 15 minutes.

Network Management

A set of procedures, equipment, and operations designed to keep a traffic network (a telephone network, for example) operating near maximum efficiency when unusual loads or equipment failures would otherwise force the network into a congested, inefficient state.

Network Management Data

A combination of data collected from the switches and data entered in the record base. This data describes the base of the network and what occurs in the network.

NFS   Network File System

A distributed-file-system protocol that allows a computer on a network to use the files and peripherals of another networked computer as if they were local.

NHR   Not Hard-to-Reach

A code (telephone number) determined to be not hard-to-reach because the attempts and failures to the code do not exceed user-defined thresholds.

NMC Network Management Center

A centralized location at the network management layer used to consolidate input from various network elements to monitor, control, and manage the state of a network in a telecommunications organization.

NOCS Network Operation Center

A group responsible for the day-to-day care of a network.

NPA Numbering Plan Area

A geographic division within which telephone directory numbers are subgrouped. A 3-digit NXX (local office) code is assigned to each NPA, where:

- N=any digit 2 through 9
- X = any digit 0 through 9

NPR

NTM Performance Reporting

NS

Number Service

NTM

Network Traffic Management

NTM **Host**

The server on which the NTM is run.

OOCC Occupancy

The time a circuit or switch is in use.

OCCH Outgoing Connections per Circuit per Hour

The outgoing peg count divided by the number of equivalent 2-way circuits.

Office

A local switch, DCC, or FEP connected to your host computer.

OFL Overflow

Number of attempts failing to find an idle circuit in a group of circuits.

One-Way Trunk

A trunk that can be seized at only one end.

Ongoing Data

Data retrieved by the `ongoing` command from the system's shared memory.

Originating Calls

Line seizures at the office.

ORR Overflow Reroute

A reroute post-hunt trunk group control option that takes the overflow traffic on a trunk group and reroutes it to a trunk group with idle capacity.

Outgoing Calls

Calls intended to complete on trunks to points outside the system (same as outgoing seizures).

Overflow Peg Count

Peg count overflowing to another trunk group or to a circuit busy signal.

OVLD Overload

An increase in offered load beyond the capacity for which the network components (for example, trunks and switching systems) are engineered.

PPage

A page is a universal resource locator (URL), part of the NTM application. A page is displayed inside a Window. The user selects, changes and transfers pages within the same window.

Parameter area

The area of a control request display that contains various control parameters.

Parameter Set

A predefined group of control parameter values that may be used to quickly apply a control to one or more switches.

PAS

Public Announcement Service

PATR   Performance and Troubleshooting Reports

This feature enables NTM personnel to collect various office and application performance data, and to output reports on request. Depending on the report type selected, the data may be real-time or hourly. The hourly data may be for a 24-hour period or less. Seven days of data are collected and stored for report access.

PC Peg Count

A count of all calls offered to a subgroup during a measurement interval.

PCI

Panel Call Indicator

PIIT Prohibit International Inbound Traffic

A reroute trunk group control option. When set to YES, does not allow inbound international traffic to be rerouted. When set to NO, allows inbound international traffic to be rerouted. See the `rr` command (4-44) in the *Input Commands Guide*.

Post-Hunt Control
A trunk group control that may affect a call that is attempting to alternate route to the next designated trunk group, for example: CANF.

PP
Preprogram

PPC
Peripheral Processor Complex

Pre-Hunt Control
A trunk group control that may affect a call before it is offered to a particular trunk group, for example: CANT, SKIP.

Preplan
Command used to create and manage pre-designated control plans to be used in emergency situations. See the `preplan` command (4-72) in the *Input Commands Guide*.

PS/UT
Pseudo-Subunit / Unit Type

PTS
Public Telecommunications Systems

QQOR
Query on Release

RRADR
Receiver Attachment Delay Readiness

RC
Routing Code

RDB
Routing Data Block

Real Time Usage
The percentage of time used out of total available real time, not including multi-task time.

Record Base
A collection of ASCII files containing reference information about the network to be managed by NTM.

Record Base Administration

The process of creating and maintaining the reference data portion of the NTM database.

Reference Data

Data that describes what the network is managing. This consists of either data about the network management center itself (such as the configuration of the center and threshold tables) or data about the network being monitored (such as the switching systems and trunk groups in the network management center's cluster). User-defined reference data is stored in the "/musr/rb" directory. Some reference data is supplied to the database by audits. This data typically changes infrequently.

Regular Expressions

A way of searching for patterns of characters in text strings. In NTM, it applies to Network Element search fields used to find particular switches or trunk groups.

Reorder Tone

A tone that is applied 120 times per minute to indicate all switching paths busy, all toll trunks busy, equipment blockages, unassigned code dialed, or incomplete registration of digits at a tandem or a toll office. Also called **Channel Busy** or **Fast Busy Tone**.

Request Page

One of the five basic types of pages used in the GUI. It is used to display control parameters before a control is applied.

Reroute

See "RR" (p. 20).

Reservation Level

The Circuit Reservation (CR) control allows the user to specify a maximum number of idle circuits to reserve and what the switch is to do with direct and/or alternate routed traffic when the reservation level is reached.

RLU

Remote Line Unit

ROA

Re-Order Announcement

Route

One or more trunk groups providing a connection between offices.

Route Group

A route group consists of one or more routes that may be used for a given destination. A route group may be accessed by more than one combination of destination and additional parameters.

RP Revertive Pulse

Revertive Pulsing is a method of signaling between switching systems in which information is conveyed from System A to System B. System B sends a sequence of pulses to System A, where the pulses are counted. System A signals System B when the correct number of pulses has been received.

RR ReRoute

An expansive trunk group control that is used to take traffic from congested or failed routes to other trunk groups not normally included in the route advance chain. These other trunk groups, called "vias," should have available idle circuits (AIC) to be used for the reroute. See the `rr` command (4-44) in the *Input Commands Guide*.

RSPTE Regional, Sectional, Primary, Toll, and End office

See the "RSPTE File" (p. 68) in the *Record Base Administration Guide*.

RSU

Remote Switching Unit

SSCCP Signaling Connection Control Part

A signaling protocol that provides additional routing and management functions for transfer of messages other than call setup between signaling points.

SCP Service Control Point

A remote database within the SS7 network that supplies the translation and routing data needed to deliver advanced network services. Also called Signal Control Point.

SDM

Supernode Data Manager

SDN Software Defined Network

A service developed for multi-location businesses that allows network managers to tailor their network to their own specific communications needs.

SDOC

Selective Dynamic Congestion Control/Automatic Congestion Control

Search Page

One of the five basic types of pages used in the GUI. It is used to request data on network elements, network connections, and controls. It can be used in simple or advanced modes.

Seizure

An attempt for a circuit in a trunk group that succeeds in obtaining a circuit.

Select Mouse Button

Mouse button used to specify an object to operate on and to manipulate objects and controls. (Usually the left mouse button.)

Set

Logical grouping of network elements (offices or trunk groups). NTM with standard features allows each office to be a member of up to four office sets, and each trunk group to be a member of up to four trunk group sets.

Shared Memory

A RAM-based data structure on the host that is used to store discrete, control, and exception data. Portion of memory accessible to multiple processes.

Signaling

The transmission of address (pulsing), supervision, or other switching information (including any information required for billing) between stations and switching systems, and between switching systems.

SILC Selective Incoming Load Control

An automatic trunk group control that can be enabled or disabled on a selected trunk group in a "From Office" when the office encounters machine congestion. See the `silc` command (4-55) in the *Input Commands Guide*.

Single File Create

The process for creating (compiling) individual record base files.

Single Office Create

The process for creating (compiling) all office-related files for one office only. A single office `create` acts directly on the current database; no `installdb` command is necessary to install the changes to the database. See the *Record Base Administration Guide*.

SKIP Skip route control

A pre-hunt trunk group control that allows all or a percentage of traffic to bypass a specific route and to advance to the next route in its normal routing pattern. See the `canf/cant/skip` command (4-13) in the *Input Commands Guide*.

SMS Service Management System

Allows provision and updating of information on subscribers and services in near-real time for billing and administrative purposes.

SQL Structured Query Language

Database language used for creating, maintaining, and viewing database data. See Chapter 3, "SQL Interpreter" in the *Data Tables Guide*.

SQL File

A data request file that lets you specify what data should be retrieved from the database or the ongoing shared memory and to define the format of the data.

SS7 Signaling System 7

Signaling protocol that uses destination routing, octet-oriented fields, variable length messages and a maximum message length allowing for 256 bytes of data. The four basic sub-protocols of SS7 are: MTP, SCCP, ISUP, and TCAP.

SSP Service Switching Point

A switch that can recognize IN (Intelligent Network) calls and route and connect them under the direction of an SCP. Also called **Signal Switching Point**.

STP Signal Transfer Point

A message switching system that permits signaling messages to be sent from one switching system to another by way of one or more other offices at which STPs are located. It reduces the number of data links required to serve a network.

STR Selective Trunk Reservation

An automatic trunk group control that reserves the last few trunks of a trunk group for critical users exclusively and eliminates the need to queue critical users for inter-switch trunks. Also called CR/TSR. See the `cr` command (4-32) in the *Input Commands Guide*.

Subnetwork

A subdivision of the network that allows parts of the network to be monitored and controlled independently of the main network.

Suffix

A user-defined string (up to 5 characters long) used to identify a particular office or trunk group. The suffix is separated from the office or trunk-group name by a hyphen.

Surveillance Data

Discrete and measurement data collected periodically from the switch.

SVC Switched Virtual Circuit

A virtual circuit connection established across a network on an as-needed basis and lasting only for the duration of the transfer.

Switch

A computer system that channels telephone calls from one place to another and keeps track of each call that it transfers.

Switch Name

A code name that identifies an office.

Syntax
The format in which a command is entered, including the input command name, parameters, and action options.

System Error
The NTM GUI presents error messages in response to conditions such as improper permission, execution errors, etc. A system error is presented when an error occurs on the NTM host during the generation of a web page or during the processing of a request from a web page (except certain control related requests).

TTandem Office
In general, an intermediate switching system for interconnecting local and toll offices. All toll offices are tandem offices. A more specific meaning of local tandem or metropolitan tandem office is an office that connects end offices to other end offices or to other tandem offices within a metropolitan area.

TCAP Transaction Capabilities Application Part
A signaling protocol that provides for transfer of non-circuit related information between signaling points.

TCU
Time Switch and Peripheral Control Unit

TDM
Time Division Multiplexing

Terminating Calls
Calls intended to complete on lines served by the system.

TFP
Transfer Prohibit

TG Trunk Group
A group of trunks with similar electrical characteristics that go between two geographical points. A trunk group performs the same function as a single trunk, except that on a trunk group multiple conversations can be carried. Trunk groups are used as traffic demands them.

Threshold
A preset limit of exceptions that each network element must exceed during each 5-minute period before NTM determines that the office is experiencing patternable trouble.

Thresholding

The process of setting values to be compared against data values (raw counts) collected from the switches every 5 minutes to determine exception conditions.

TID

Terminal Identifier

To Office

Internal or external network element that is the termination of a trunk group.

TPC

Telephony Processor Complex

Traffic Network

An arrangement of channels, such as loops and trunks, associated switching arrangements, and station equipment designed to handle a specific body of traffic; a subset of the facility network.

Trunk

A telephone communication path or channel between two points, one of them usually being a telephone company central office or switching center.

Trunk Group

See “TG” (p. 23).

Trunk Group Number

Number assigned to a trunk group in the switch.

TSG

Trunk Subgroup

TTO

Transmitter Time-Out

Two-Way Trunk

A trunk that can be seized at either end.

UUDTS

Unitdata Services

URW User Report Writer

The User Report Writer consists of the transaction processing system report writer software package and a system command set. The transaction processing system generates informational reports based on data that changes periodically.

Usage

A measure of trunk or equipment occupancy expressed in Erlangs or CCS.

VVacant Code

An unassigned numbering plan area, central office, or station code. A call placed to a vacant code is normally directed to a VCA (vacant code announcement).

Validate

A command used to verify that the values and actions specified are correct for a specific display or page.

VB

V-B (terminating) trunk group. A trunk group that connects a via office (V) to a terminating office (B). See "AB" (p. 1) and "AV" (p. 3).

Via Office

An office that transits a rerouted call between the originating office and the terminating office.

Via Trunk Group

A trunk group designated to carry the calls redirected by a reroute control activated on the "reroute from" trunk group of the reroute control. If a trunk group is identified as a "via trunk group" it is the "AV" portion of the "AV"-"VB" path for rerouted calls.

VRTO Via Route Turnoff Override

VRT is a reroute option that protects regular traffic from rerouted traffic, by not allowing rerouted traffic to use a via TG that is filling with regular traffic. VRTO overrides the VRT option so that network managers can use the via trunk group anyway. See the `rr` command (4-44) in the *Input Commands Guide*.

WWindow

A window is box-type graphic displayed when specific buttons, icons, function keys or hot keys are selected in a windows operating system environment. Each window contains various control attributes including a means to close the box, typically an "X" in the upper right corner. The window identifier is displayed in the task bar. The user opens and closes windows.

Index