

HPE NonStop™ ASAP Server Manual

Abstract

HPE NonStop™ Availability Statistics and Performance (ASAP) is an availability, state, and performance statistics collection infrastructure for the HPE NonStop™ operating system and application resources.

Product Version

ASAP SE30V3.2

Supported Release Version Updates (RVUs)

This manual supports D42.00, G06.00, H06.03, J06.03, L15.02 and all subsequent D-series RVUs, G-series RVUs, H-Series, J-Series and L-Series RVUs until otherwise indicated in a new edition.

<u>Part Number</u>	<u>Published</u>
52303-010	March 2010
P11256-001	October 2018

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Export of the information contained in this publication may require authorization from the U.S. Department of Commerce.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgements

Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation. Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Java and Oracle are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Intel®, Itanium®, Pentium®, Intel Inside®, and the Intel Inside logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Document History

Part Number	Product Version	Published
522303-002	ASAP SE30V2	October 2003
522303-004	ASAP SE30V2	January 2004
522303-005	ASAP SE30V2.5	October 2004
522303-006	ASAP SE30V2.7	May 2006
522303-007	ASAP SE30V2.8	January 2007
522303-008	ASAP SE30V3.0	April 2008
522303-009	ASAP SE30V3.1	December 2008
522303-010	ASAP SE30V3.2	March 2010
P11256-001	ASAP SE30V3.2	October 2018

Table of Contents

HPE NonStop™ ASAP Server Manual	1
Abstract	1
Product Version	1
Supported Release Version Updates (RVUs).....	1
Table of Contents	3
List of Figures and Tables	9
What's New in This Manual	10
New and Changed Information.....	10
Previous Changes.....	10
About This Manual	20
Audience	21
Manual Organization	21
Related Resources.....	22
Notation Conventions.....	22
Hypertext Links.....	22
General Syntax Notation	22
Notation for Messages.....	26
Notation for Management Programming Interfaces.....	28
Section 1: Introduction to Availability Statistics and Performance (ASAP)	29
ASAP Client Features	29
ASAP Server Features.....	30
ASAP Extension Features	31
ASAP Extension Architecture	32
ASAP Hybrid Features	32
Data Collection and Monitoring	33
Availability Overview	34
Components of ASAP	36
Presentation Interfaces	37
Conversational Interface.....	38
Graphical User Interface.....	38
Batch Reporting.....	38
Command and Control	39
Database	39
Collection.....	40
Node Monitors	40
Statistics Gathering Processes.....	40
Application Statistics Gathering Process	40
CIP Statistics Gathering Process.....	40
COMM Statistics Gathering Process	41
CPU Statistics Gathering Process.....	41
Disk Statistics Gathering Process.....	41
Expand Statistics Gathering Process	41

File Statistics Gathering Process.....	42
Node Statistics Gathering Process.....	42
Process Statistics Gathering Process.....	42
RDF Statistics Gathering Process.....	42
Spooler Statistics Gathering Process.....	43
SWAP Statistics Gathering Process.....	43
Tape Statistics Gathering Process.....	43
TCP/IP Statistics Gathering Process.....	43
TMF Statistics Gathering Process.....	44
CSG.....	44
SSG.....	44
COLLECT/ASAPDB.....	44
Section 2: Installing ASAP on the Server	46
Task 1: Ensure Your System Meets Requirements.....	46
Hardware.....	46
Terminals and Terminal Emulators.....	46
Software.....	46
Task 2: Install ASAP Server Product Files.....	47
Task 3: Set Up ASAP Server.....	48
Full Install on This Node.....	49
Remote Monitor Install.....	53
Task 4: Configure ASAP Components.....	58
Confirm ASAP Configuration Settings.....	58
ASAP Object Files.....	62
Size the ASAP Database Subvolume.....	69
Recommended Sample Rates.....	71
Starting and Stopping ASAP.....	71
Starting ASAP.....	72
Stopping ASAP.....	74
Using OBEY Files.....	75
Section 3: Running ASAP.....	76
Starting the ASAP Collector.....	76
Starting the ASAP Monitors and SGPs.....	78
Starting the Monitor Processes.....	80
Getting Collector and Monitor Status Information.....	80
Starting ASAP With the Default Configuration.....	82
Changing the ASAP Configuration Dynamically.....	82
Viewing Statistics.....	82
Shutting Down the ASAP Collector and ASAP Monitors.....	83
Synchronizing the Time of Day on Selected Nodes.....	83
Monitoring Different Sets of Network Nodes.....	84
Example A.....	85
Example B.....	85
Viewing the ASAP Event Log.....	85
ASAP Help.....	86
Section 4: Using Discrete Object Thresholds (DOTs).....	88

DOTs Overview.....	88
Specifying Domains and Objectives	92
The Objectives Database	92
MONITOR Usage Considerations.....	95
GOAL and RANK Usage Considerations.....	96
EMS Event Considerations	97
Status Change Events.....	98
Event Considerations Summary	99
System Entity Specifics.....	100
CIPCLIM Entity-Supported Objective Attributes	101
CIPMONITOR Entity-Supported Objective Attributes	103
CIPPROVIDER Entity-Supported Objective Attributes	104
COMM Entity-Supported Objective Attributes.....	104
CPU Entity-Supported Objective Attributes	105
DISK Entity-Supported Objective Attributes	105
EXPAND Entity-Supported Objective Attributes	106
EXPANDIP Entity-Supported Objective Attributes.....	107
FILE Entity-Supported Objective Attributes	107
HYBRID Entity-Supported Objective Attributes	108
PROCESS Entity-Supported Objective Attributes	108
PROCESSBUSY Entity-Supported Objective Attributes.....	109
RDF Entity-Supported Objective Attributes	110
SPOOLER Entity-Supported Objective Attributes	110
SWAP Entity-Supported Objective Attributes	110
SYSTEM Entity-Supported Objective Attributes	111
TAPE Entity-Supported Objective Attributes	112
TCP/IP and Telserv Entities-Supported Objective Attributes.....	112
TMF Entity-Supported Objective Attributes	128
Section 5: File and Process Monitoring.....	130
Specifying Wildcard Names	130
Monitoring Processes from an Object File.....	131
Monitoring Files and Processes Using OSS Pathnames	132
Logical Grouping.....	133
Domain Aggregation	134
Creating Specific Aggregate Domains (Recommended)	134
Setting Global Aggregates for All Domains	134
Aggregate Only Records.....	134
Creating Specific Aggregate-Only Domains (Recommended).....	135
Setting Global Aggregate Only for All Domains	135
Aggregation Defined	135
State Propagation	135
Aggregation Rules.....	136
Attribute Propagation	137
Status and OpState.....	137
Attributes without States	138
Process PState and WState Attributes.....	138

Default Process MetricRule Settings.....	139
Modifying the Default Aggregation Settings	140
Section 6: ASAP Conversational Interface Commands	141
Considerations	143
ACTION Command.....	144
APP Command	147
CIP Command	155
Examples.....	157
CLEANUP Command.....	157
Example	158
COMM Command	158
COMM Statistics.....	162
Examples.....	163
COMMIT Command.....	163
CPU Command.....	163
CPU Statistics	166
Examples.....	168
DB Command.....	169
Examples.....	170
DISK Command	171
Examples.....	177
EDL Command	178
Examples.....	178
EXIT Command	179
FC Command.....	179
FILE Command.....	179
Considerations	182
Examples.....	182
GOAL Command	184
Considerations	193
Examples.....	194
HELP Command	198
Example	199
LH Command.....	199
LH Statistics	204
Examples.....	206
LOG Command.....	208
Example	208
MONITOR Command.....	208
Considerations	216
Examples.....	217
NODE Command	221
NODE Statistics.....	223
Examples.....	223
OBEY Command.....	224
Example	224

PAUSE Command	224
Example	225
PROCESS Command	225
Examples.....	227
PROCESSBUSY Command	228
PROCESSBUSY Statistics.....	231
Examples.....	232
RANK Command.....	233
Considerations	242
Examples.....	243
RDF Command	247
Examples.....	248
SET Command	251
SHOW Command	292
Examples.....	293
SHUTDOWN Command	293
Examples.....	294
Considerations	294
SPOOLER Command	294
STARTUP Command.....	299
Example	300
Considerations	300
STATUS Command	301
Example - Status Monitor	304
Example - Status Collect	304
Example - Status Actions	305
SWAP Command	306
SWAP Statistics.....	308
Examples.....	310
SYNC Command.....	311
SYSTEM Command.....	311
Example	312
TAPE Command	312
Consideration	313
Examples.....	313
TCP Command	315
Examples.....	317
TMF Command.....	318
Consideration	319
Examples.....	319
VOLUME Command	322
Example	322
Appendix A: Querying the ASAP Database.....	323
Standard Enform Query	323
ASAPCOL as an Enform Server	324
Appendix B: ASAP Data Definitions.....	326

DDL Section Definitions	326
Primary Key Definitions	326
Event Definitions.....	327
Query Key Definitions.....	327
Extended Primary Key.....	327
Extended Query Key	328
Application Entity Statistics Definitions	328
CIP Statistics Definitions	328
Comm Statistics Definitions.....	334
CPU Statistics Definitions.....	335
Disk Entity Statistics Definitions.....	336
Expand Entity Statistics Definitions	338
File Entity Statistics Definitions.....	340
Node Entity Statistics Definitions.....	341
Process Statistics Definitions	341
ProcessBusy Statistics Definitions.....	343
RDF Statistics Definitions	344
Spooler Statistics Definitions	344
SWAP Statistics Definitions.....	346
Tape Statistics Definitions	347
TCP/IP Entity Statistics.....	348
TMF Entity Statistics Definitions	371
DDL Section Record Declarations.....	372
ASAP Database Record Declarations	372
Enform Server Declarations.....	372
Appendix C: ASAP CI DOTs Warning Messages	374

List of Figures and Tables

- Figure 1-1. ASAP Components.....37
- Figure 2-1. ASPSETUP Main Menu, AsapInstall49
- Figure 2-2. Installation Configuration for Full Install on This Node.....50
- Figure 2-3. Input Help for the ASAP Install subvolume Field51
- Figure 2-4. First Page of Output From the START FULL INSTALL Function.....52
- Figure 2-5. Completion Display From the START FULL INSTALL Function.....53
- Figure 2-6. ASAP Help Text for Remote Monitor Install.....54
- Figure 2-7. Remote Monitor Install Display With User Entries55
- Figure 2-8. Selecting Remote Install Nodes for Remote Installation56
- Figure 2-9. Output From Remote Install.....57
- Figure 2-10. Sample Remote Install Completion Screen58
- Table 3-1. STATUS COLLECTOR Output.....81
- Figure 3-1. Monitoring Different Sets of Network Nodes.....84
- Table 6-1. ASAP Command Summary142
- Table 6-2. COMM Statistics and Descriptions162
- Table 6-3. CPU Statistics and Descriptions166
- Table 6-4. CPU Statistics Favorable for System Performance.....167
- Table 6-5. DISK Statistics and Descriptions176
- Table 6-6. LH Statistics and Options204
- Table 6-7. Statistic Counters.....205
- Table 6-8. Node Statistics and Options223
- Table 6-9. Processbusy Statistics and Options.....231
- Table 6-10. STATUS Command Output302
- Table 6-11. SWAP Statistics and Descriptions309

What's New in This Manual

New and Changed Information

The ASAP 3.2 version of this manual contains the following changes:

- Added escalating and Boolean goals and new action tokens to the [DOTS Overview](#) section.
- Added Boolean goal names and new action tokens to the [ACTION Command](#) section.
- Added escalating and Boolean goal information to the [GOAL Command](#) section.
- Added escalating and Boolean goal information to the [RANK Command](#) section.
- Added a new PLUS option to the [SET SPOOLER](#) command for high performance Spooler environments.
- Added a NOTE to [SET OBJECTIVESEVENTSTATE](#) and slightly changed the description of the option.
- Added [Appendix C](#) to describe DOTs related warning messages.

Previous Changes

The ASAP 3.1 version of this manual contains these changes:

- The Cluster I/O Protocol (**CIP**) entity has been added for use when ASAP is run on NonStop™ servers using Cluster I/O Module (CLIM) devices. For more information see the following sections:
 - [CIP Statistics Gathering Process](#)
 - [CIP Entity – Supported Objective Attributes](#)
 - [CIP Command](#)
 - [CIP Entity Statistics](#)
- The **CPU** entity has added a new IPU attribute for use when ASAP is run on NonStop™ Blades systems.

- [CPU Entity - Supported Objective Attributes](#)
- [CPU Entity Statistics](#)
- The **PROCESS** entity has added new attributes for use when ASAP is run on NonStop™ Blades systems.
 - [Process Entity - Supported Objective Attributes](#)
 - [PROCESS Entity Statistics](#)
- The **PROCESSBUSY** entity has added new attributes for use when ASAP is run on NonStop™ Blades systems.
 - [PROCESSBUSY - Supported Entity Attributes](#)
 - [PROCESSBUSY Entity Statistics](#)
- Minor corrections throughout the manual.

The eight edition (ASAP 3.0 version) of this manual contains these changes:

- The **TCP/IP** and **Telserv** entities have undergone major revisions. For more information, see the following sections:
 - [TCP/IP Statistics Gathering Process](#)
 - [TCP/IP and Telserv Entities-Supported Objective Attributes](#)
 - [TCP Command](#)
 - [TCP/IP Entity Statistics](#)
- A new **Swap SGP** is being introduced in this release. The Swap SGP collects availability and performance information about the KMSF subsystem. For more information, see the following sections:
 - [SWAP Statistics Gathering Process](#)
 - [SWAP Entity-Supported Objective Attributes](#)
 - [SWAP Command](#)

- [SWAP Statistics Definitions](#)
- The **CPU** entity has 6 new memory statistics including memory size, swappable pages, free pages, locked pages, maximum memory that can be locked, and the highest amount of memory locked since the CPU was coldloaded. For more information see [CPU Statistics](#).
- A new **SET PARTITION <entity> KEY | RELATIVE** options have been added to allow different historical database performance characteristics. You can now define a database table to be a Relative file instead of a Key-sequenced file, and you can control the refresh EOF setting. See the [SET Command PARTITION](#) option for more information.
- The Query command has been removed along with other references to the Query SGP and its data.

The seventh edition (ASAP 2.8 version) of this manual contains these changes:

- A new TCP command was added. For more information, see the following sections:
 - [TCP/IP Statistics Gathering Process](#)
 - [TCP/IP and Telserv Entities-Supported Objective Attributes](#)
 - [TCP Command](#)
 - [TCP/IP Entity Statistics](#)
- A new Query command was added. For more information, see the following sections:
 - **Query Statistics Gathering Process (deleted)**
 - **Query Command (deleted)**
- A new SET TCP ON|OFF option has been added to enable the TCP entity. For more information, see the [SET Command](#).
- A new SET QUERY ON|OFF option was added to enable the Query entity. For more information, see the Query entity option of the [SET Command](#).

The sixth edition (ASAP 2.7 version) of this manual contains these changes:

Added usage notes to:

- [Monitoring Processes from an Object File](#)
- [Monitoring Files and Processes Using OSS Pathnames](#)
- The *pathname* and *\$process* parameters for the [MONITOR Command](#)

The fifth edition of this manual contains these changes:

- A SET ACTION option allows you to configure ASAP to take actions when user-defined goals are not met. For more information, see the [SET Command](#).
- A SET DBMAXDAYS option allows you to configure ASAP so that it automatically retains days, weeks, months, or years of long-term database history. For more information, see the [SET Command](#).
- A DB command displays detailed information about disk space utilization for ASAP database files when the SET DBMAXDAYS option is enabled. For more information, see the [DB Command](#).
- A SET DBROLLOVERSUBVOLPREFIX option allows you to configure the ASAP database archive subvolume.
- A SET NODEDOWNALERT option allows you to configure ASAP to generate a Node Down alert event if a NonStop™ server node goes down or becomes inaccessible from an ASAP Collector. For more information, see the [SET Command](#).
- A Notify capability allows you to automatically send e-mail, phone, pager alerts and reports. You can send notifications in either color-coded HTML format for use with e-mail or internet servers, or as plain text messages for use with wireless phones or pagers. For more information, see the ASAP Client Manual.
- You can now secure clients settings. A Set Password button has been added to the General tab of the ASAP Client Properties window. This button displays a dialog box that allows you to protect various property sheets with a password (and lock functions) in the ASAP environment. For more information, see the ASAP Client Manual.
- Client/Server EDL version checking is now performed automatically. A new EDL synchronization dialog box automatically appears whenever an EDL version mismatch occurs, allowing automatic EDL version synchronization. For more information, see the ASAP Client Manual.
- An Auto Restart Session check box has been added to the SSG tab of the ASAP Properties window. When you select it, the ASAP client now automatically restarts failed host ASAP Server sessions; for example, in the event of a CPU failure. For more detail, see Auto Restart Session check box on the ASAP Client SSG property sheet.
- A new smart plug-in for OpenView Operations is included on the ASAP Client CD.

- A process for gathering communication line statistics lets you collect availability information for SWAN communication lines. For additional information, see the SET COMM, and COMM commands.
- Specify which CPU the SGPs will execute. See the SET *entity* command.
- A Timeout option for the Status Monitor command lets you specify the elapsed time for an ASAP SGP to reply. The Status command displays entity error information in the unlikely event an SGP is in an error state or an SGP cannot start. For additional information, see the [STATUS Command](#).
- The ASAP monitor has been enhanced to restart SGPs at a 15-minute interval when an SGP is accidentally stopped or is in an error state.
- ACTION, a parameter to the [RANK Command](#), lets you define a specific recovery action to take when the objective is not met.
- A new command, the [ACTION Command](#), lets you define and list stored actions and retrieve the stored action for a specific entity-domain-attribute-operator combination.
- Process records contain two attributes in Process records. Count is the number of domains. ACount is the number of attributes that are alerting when viewed at the detail level, and is the number of domains that are alerting when viewed at the aggregate level.
- You can:
 - Monitor processes by their Guardian object filename or OSS pathname.
 - Use wildcard process names to specify the processes to monitor.
 - Logically group files and subvolumes into hierarchical groups.
 - Aggregate data across any level of logical groupings of files.
 - Write only aggregate file records to the historical database.
 - Use wild-card file names to specify files to be monitored.
 - Monitor OSS files by specifying the OSS pathname.
 - Specify custom event numbers for specific objectives using the RANK command. Custom event numbers in the range 6000 through 9999 are available for this purpose.
- The OSS Permissions attribute has been added to file records.

The fourth edition of this manual contained these changes:

- The only difference between 522303-002 and 522303-004 is the correct display of Section 4.
- Updated Process Statistics Gathering Process
- Added a new object file in The ASAPX Library
- Updated Starting the Monitor Processes
- Updated values in:
 - [DISK Entity-Supported Objective Attributes](#)
 - [EXPAND Entity-Supported Objective Attributes](#)
 - [EXPANDIP Entity-Supported Objective Attributes](#)
 - [FILE Entity-Supported Objective Attributes](#)
 - [PROCESS Entity-Supported Objective Attributes](#)
 - [RDF Entity-Supported Objective Attributes](#)
 - [SPOOLER Entity-Supported Objective Attributes](#)
 - [TAPE Entity-Supported Objective Attributes](#)
 - [TMF Entity-Supported Objective Attributes](#)
- Added options and examples to the APP Command
- Added examples to the MONITOR Command
- Added options and examples to the PROCESS Command
- Added options and examples to the SET Command
- Added Section 5: File and Process Monitoring
- Updated Process Statistics Definitions

The third edition of this manual contained these changes:

- Updated the formula in Size the ASAP Database Subvolume
- Added GOAL and RANK Usage Considerations to further explain how to use the

RANK command.

- Added [EMS Event Considerations](#) to explain the options you have in ASAP for working with EMS events.
- Added the STATUS and NOEMS options to the *rankspec* parameter of the RANK command. Use these new options to change default behavior of system entities and to turn off event generation for status changes in ASAP system objects, respectively.

The second edition of this manual contained these changes:

- Added these new object files:
 - [ASAPEXP Object File](#)
 - [ASAPFIL Object File](#)
 - [ASAPPRO Object File](#)
 - [ASAPRDF Object File](#)
 - [ASAPSPL Object File](#)
 - [ASAPTAP Object File](#)
 - [ASAPTMF Object File](#)
- Added these ASAP components:
 - [File Statistics Gathering Process](#)
 - [Process Statistics Gathering Process](#)
 - [RDF Statistics Gathering Process](#)
 - [Spooler Statistics Gathering Process](#)
 - [Tape Statistics Gathering Process](#)
 - [TCP/IP Statistics Gathering Process](#)
- Added these new server commands:
 - [COMMIT Command](#)
 - [EDL Command](#)
 - [FILE Command](#)
 - [LOG Command](#) (replaces INFO)

- [MONITOR Command](#)
- [PROCESS Command](#)

Note

The PROCESS command in ASAP Release 1 has become PROCESSBUSY in ASAP Release 2. The Selected Process function that PROCESS now performs is new to ASAP Release 2.

- [PROCESSBUSY Command](#)
 - **QUERY Command (deleted)**
 - [RDF Command](#)
 - [SHOW Command](#)
 - [SPOOLER Command](#)
 - [TAPE Command](#)
 - [TMF Command](#)
- Added these new options to the LH Command
- IP
 - LINE
 - PATH
 - STATE
- Added these new options to the SET Command
- EDL
 - FILE
 - LOGINTERVAL
 - LOGSUPPRESS
 - OBJECTIVESAUDIT
 - OBJECTIVESDB
 - OBJECTIVESEVENT
 - OBJECTIVESEVENTUP
 - OBJECTIVESEVENTCONSOLE

- OBJECTIVESEVENTCONSOLEUP
 - OBJECTIVESEVENTSTATE
 - OBJECTIVESEVENTMAX
 - OBJECTIVESRANK
 - PARTITION
 - PROCESS
 - SPOOLER
 - TAPE
 - TMF
- For File, Process, RDF, Spooler, Tape, TMF entities:
- Added statistics definitions in [DDL Section Definitions](#)
 - Updated [DDL Section Record Declarations](#)

About This Manual

This manual describes these Availability Statistics and Performance (ASAP) products:

- The ASAP Server monitors, gathers, and analyzes system availability throughout a network of NonStop™ servers. System Entities are CIP, CPU, Comm, Disk, Expand, File, Hybrid¹, Node, Process, Processbusy, RDF, Spooler, Swap, System, Tape, TCP/IP, Telserv², and TMF.

The ASAP Extension provides an Application Programmer Interface (API) so you can monitor the availability and performance of your application domains. The API lets you fully integrate Application domain statistics with ASAP Client and Server functions.

ASAP Hybrid extends the Application Programmer Interface (API) to applications running on the Linux operating system. It allows you to monitor the availability and performance of applications that span NonStop™ operating system and Linux systems.

- The ASAP Client provides both graphical user interface (GUI) and intelligent agent functions. It displays NonStop™ OS performance and entity state information, as well as customer application availability.

You can use the ASAP Client to view and analyze real-time or historical network-wide views of Application, CIP, Comm, CPU, Disk, Expand, File, Hybrid, Node, Process (user selected), Processbusy, RDF, Spooler, Swap, System, Tape, TCP/IP, Telserv, and TMF entities. Informational views can encompass all of your NonStop™ OS nodes, reflect a selected set of nodes, or focus on a single node, entity, or object. The ASAP Client also has built-in agent functions that automatically retrieve and analyze availability information from the ASAP Server. For more information, see the ASAP Client Manual.

- The ASAP Extension (ASAPX) is used to collect, measure, view, and analyze application service-level metrics to track the productivity, performance, and availability of applications.

These service-level metrics are automatically evaluated against predefined objectives to establish alert priorities so that you can quickly identify when objectives are not being met for an application process. For more information, see the ASAP Extension Manual.

- ASAP Hybrid is used to collect, measure, view, and analyze application service-level metrics to track the productivity, performance, and availability of Linux-based applications and applications that span NonStop™ and Linux systems.

¹ The Hybrid entity is an optional part of the ASAP Hybrid product.

² The TCP/IP and Telserv entities are provided in the optional ASAP TCP/IP Plug-in.

These service-level metrics are automatically evaluated against predefined objectives to establish alert priorities so that you can quickly identify when objectives are not being met for an application process. For more information, see the *ASAP Hybrid Manual*.

This manual also describes these important tools for your ASAP environment:

- The Entity Definition Language (EDL) lets you describe an entity and its associated data attributes required for the ASAP Client, Server, and Extension.

The system entities APP, CIP, Comm, CPU, Disk, Expand, File, Hybrid, Node, Process, ProcessBusy, RDF, Spooler, Swap, System, Tape, TCP/IP, Telserv, and TMF have entity definitions in the ASAP3SYS file. This file must not be altered without the consent of HPE. Customer-developed applications in the ASAP environment use the EDL for defining application metrics and display formats. For more information on creating custom EDLs for user applications, see the ASAP Extension Manual and *ASAP Hybrid Manual*.

- Discrete Object Thresholds (DOTs) let you set discrete objectives against specific attributes for any ASAP entity.

DOTs provides a database in ASAP to store and retrieve domain names and objective values; a user command interface; an event generation service; and an API to calculate ASAP (OEM) states and retrieve information from the database. DOTs require that you define entities using EDL and that entity information is available for each caller of the DOTs API.

Audience

The manual is intended for system operators, system administrators, and others who need information about NonStop™ operating system performance, entity states, and user application availability information.

Manual Organization

Section/Appendix	Description
1	Introduction to Availability Statistics and Performance (ASAP)
2	Section 2: Installing ASAP on the Server
3	Section 3: Running ASAP
4	Section 4: Using Discrete Object Thresholds (DOTs)
5	Section 5: File and Process Monitoring
6	Section 6: ASAP Conversational Interface Commands
A	Appendix A: Querying the ASAP Database
B	Appendix B: ASAP Data Definitions

Related Resources

This manual describes the ASAP Client and ASAP Server, the ASAP base components. For details on the ASAP Client, see the *ASAP Client Manual*.

For details on event messages generated by ASAP, see the *ASAP Messages Manual*.

The ASAP Extension (ASAPX) is a related product that is used to collect, measure, view, and analyze application service-level metrics. This manual briefly discusses the architecture, installation, components, and functions of ASAPX. For details on ASAPX, see the ASAP Extension Manual.

Additional technical resources including frequently asked questions, example screen shots, softdoc and readme file history, database calculators, Java native interface, and shared wrappers are available at the HPE NonStop™ ASAP technical portal <http://NonStopAsap.com>.

ASAP Hybrid (ASAPH) is a related product that is used to collect, measure, view, and analyze application service-level metrics from Linux systems. This manual briefly discusses the functions of ASAPH. For details on ASAPH, see the *ASAP Hybrid Manual*.

Notation Conventions

Hypertext Links

Blue underline is used to indicate a hypertext link within text. By clicking a passage of text with a blue underline, you are taken to the location described. For example:

This requirement is described under [Backup DAM Volumes and Physical Disk Drives](#) on page 3-2.

General Syntax Notation

This list summarizes the notation conventions for syntax presentation in this manual.

UPPERCASE LETTERS

Uppercase letters indicate keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

MAXATTACH

lowercase italic letters

Lowercase italic letters indicate variable items that you supply. Items not enclosed in brackets are required. For example:

file-name

computer type

Computer type letters within text indicate C and Open System Services (OSS) keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

myfile.c

italic computer type

Italic computer type letters within text indicate C and Open System Services (OSS) variable items that you supply. Items not enclosed in brackets are required. For example:

pathname

[] Brackets

Brackets enclose optional syntax items. For example:

TERM [\system-name.] \$terminal-name

INT[ERRUPTS]

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

FC [num]
 [-num]
 [text]

K [X | D] address

{ } Braces

A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
LISTOPENS PROCESS { $appl-mgr-name }
                  { $process-name   }

ALLOWSU { ON | OFF }
```

| Vertical Line

A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
INSPECT { OFF | ON | SAVEABEND }
```

... Ellipsis

An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

```
M address [ , new-value ]...
[ - ] { 0|1|2|3|4|5|6|7|8|9 }...
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
"s-char..."
```

Punctuation

Parentheses, commas, semicolons, and other symbols not previously described must be typed as shown. For example:

```
error := NEXTFILENAME ( file-name ) ;

LISTOPENS SU $process-name.#su-name
```

Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must type as shown. For example:

```
"[ repetition-constant-list ]"
```

Item Spacing

Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
CALL STEPMOM ( process-id ) ;
```

If there is no space between two items, spaces are not permitted. In this example, no spaces are permitted between the period and any other items:

```
$process-name.#su-name
```

Line Spacing

If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
ALTER [ / OUT file-spec / ] LINE  
  
    [ , attribute-spec ]...
```

!i and !o

In procedure calls, the !i notation follows an input parameter (one that passes data to the called procedure); the !o notation follows an output parameter (one that returns data to the calling program). For example:

```
CALL CHECKRESIZESEGMENT ( segment-id           !i  
                        , error                 ) ;   !o
```

!i,o

In procedure calls, the !i,o notation follows an input/output parameter (one that both passes data to the called procedure and returns data to the calling program). For example:

```
error := COMPRESSEEDIT ( filenum ) ;           !i,o
```

!i:i

In procedure calls, the !i:i notation follows an input string parameter that has a corresponding parameter specifying the length of the string in bytes. For example:

```
error := FILENAME_COMPARE_ ( filename1:length    !i:i  
                          , filename2:length ) ;   !i:i
```

!o:i

In procedure calls, the !o:i notation follows an output buffer parameter that has a corresponding input parameter specifying the maximum length of the output buffer in bytes. For example:

```
error := FILE_GETINFO_ ( filenum           !i  
                        , [ filename:maxlen ] ) ;      !o:i
```

Notation for Messages

This list summarizes the notation conventions for the presentation of displayed messages in this manual.

Bold Text

Bold text in an example indicates user input typed at the terminal. For example:

```
ENTER RUN CODE
```

```
?123
```

```
CODE RECEIVED:      123.00
```

The user must press the Return key after typing the input.

Nonitalic text

Nonitalic letters, numbers, and punctuation indicate text that is displayed or returned exactly as shown. For example:

```
Backup Up.
```

lowercase italic letters

Lowercase italic letters indicate variable items whose values are displayed or returned. For example:

```
p-register
```

```
process-name
```

[] Brackets

Brackets enclose items that are sometimes, but not always, displayed. For example:

```
Event number = number [ Subject = first-subject-value ]
```

A group of items enclosed in brackets is a list of all possible items that can be displayed, of which one or none might actually be displayed. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or

horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
proc-name trapped [ in SQL | in SQL file system ]
```

{ } Braces

A group of items enclosed in braces is a list of all possible items that can be displayed, of which one is actually displayed. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
obj-type obj-name state changed to state, caused by  
{ Object | Operator | Service }  
  
process-name State changed from old-objstate to objstate  
{ Operator Request. }  
{ Unknown. }
```

| Vertical Line

A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
Transfer status: { OK | Failed }
```

% Percent Sign

A percent sign precedes a number that is not in decimal notation. The % notation precedes an octal number. The %B notation precedes a binary number. The %H notation precedes a hexadecimal number. For example:

```
%005400
```

```
%B101111
```

```
%H2F
```

```
P=%p-register E=%e-register
```

Notation for Management Programming Interfaces

This list summarizes the notation conventions used in the boxed descriptions of programmatic commands, event messages, and error lists in this manual.

UPPERCASE LETTERS

Uppercase letters indicate names from definition files. Type these names exactly as shown. For example:

```
ZCOM-TKN-SUBJ-SERV
```

lowercase letters

Words in lowercase letters are words that are part of the notation, including Data Definition Language (DDL) keywords. For example:

```
token-type
```

!r

The !r notation following a token or field name indicates that the token or field is required. For example:

```
ZCOM-TKN-OBJNAME          token-type ZSPI-TYP-STRING.          !r
```

!o

The !o notation following a token or field name indicates that the token or field is optional. For example:

```
ZSPI-TKN-MANAGER          token-type ZSPI-TYP-FNAME32.          !o
```

Section 1: Introduction to Availability Statistics and Performance (ASAP)

The Availability Statistics and Performance (ASAP) software product provides an availability monitoring infrastructure for monitoring the availability and performance of:

- System object information—operational status and performance of NonStop™ systems
- Application information—user-defined application domain availability statistics

The Entity Definition Language (EDL) lets you describe an entity and its associated data attributes, which is required for using the ASAP Client, Server, and Extension.

The system entities CIP, Communication, CPU, Processbusy, Disk, Expand, File, Hybrid, Node, Process, RDF, Spooler, Swap, System, Tape, TCP/IP, Telserv, and TMF are static system entities and should not be modified without the consent of HPE.

Customer-developed applications in the ASAP environment use EDL for defining application metrics and display formats. Only application developers should add entity definitions to the custom EDL files for applications in the ASAP environment. For information on creating custom EDL files, see the ASAP Extension Manual and *ASAP Hybrid Manual*.

Discrete object thresholds (DOTs) let you set discrete objectives against specific attributes for any ASAP entity. ASAP provides a DOTs database for storing and retrieving the entities and associated objective values. DOTs provides a configurable event generation service.

ASAP Client Features

- Windows look and feel
- Runs on Microsoft Windows NT, Windows 95, Windows 98, Windows 2000, and Windows ME
- Localized high-speed statistics object cache
- 3-D graphical views of all ASAP statistical entities
- Real-time, network-wide State and Performance views

- Browse windows for detailed entity analysis and reporting
- Graph windows for multiple real-time views of host objects
- Report windows for customized reporting of selected objects
- Automatic host communication using the Client Server Gateway
- Performance and State icons propagated upward through tree view
- Hierarchical tree view of nodes, entities, objects, and applications
- Context-sensitive menus on graphs and grids allow data mining
- Encapsulated interface to Open Enterprise Management (OEM) Gateway
- Reporting of App, CIP, Comm, CPU, Disk, Expand, File, Node, Process, ProcessBusy, RDF, Spooler, Swap, System, Tape, TCP/IP, and TMF entities
- Full customization of displays, including entity and attribute specification
- Object State Change (OSC) window lets you get a history of object state changes

ASAP Server Features

- Enables network-wide availability monitoring
- Monitors performance and availability objectives
- Synchronizes system time clocks
- Allows monitoring from a single location
- Uses but does not depend on Measure
- Has very low availability collection overhead
- Includes user-defined discrete object threshold EMS events
- Forms the core availability infrastructure for the HPE NonStop™ operating system

ASAP Server Architecture

- NonStop™ servers and collectors
- Domain registration through the API Extension
- Support for multiple node availability infrastructure
- Microsoft Windows Client interface
- Support for open App, CIP, Comm, CPU, Disk, Expand, File, Node, Process, ProcessBusy, RDF, Spooler, Swap, System, Tape, TCP/IP, and TMF entities

ASAP Server Database

- Automatically maintained
- Low disk space utilization
- Requires no operator intervention
- Automatic archival of Object History
- Provides network-wide performance and state history
- Normalized, allowing fast, ad hoc, real-time queries
- Can partition database across multiple disk volumes

ASAP Extension Features

- Easy-to-use API
- Domains of service
- Availability monitoring
- Metric calculation and generation
- Alerts based on user-defined objectives
- Data normalized in the ASAP Server database

- Full integration with the ASAP Client and Server

ASAP Extension Architecture

- Monitors and Statistics Gathering processes (SGPs)
- Nonmessaging data collection
- Shared memory segment counters
- Application Programmer Interface
- Multilevel domain name registration
- Synchronized sampling across all ASAP Server entities

ASAP Extension Discrete Object Thresholds (DOTs) Features

- Objective thresholding based on domain names and levels of the domain name
- Thresholds based on the DOTs operators $>$, $<$, $=$, $<>$, $<=$, and $>=$ on any application metric
- Objectives thresholds and alerts reported by the ASAP Client based on the actual threshold
- User configurable EMS threshold events

ASAP Hybrid Features

- Easy-to-use Linux API
- Domains of service
- Availability monitoring
- Metric calculation and generation
- Alerts based on user-defined objectives

- Data normalized in the ASAP Server database on a NonStop™ server
- Full integration with the ASAP Client and Server

ASAP Hybrid Architecture

- Monitors and Statistics Gathering agent
- Non-messaging data collection
- Shared memory segment counters
- Application Programmer Interface
- Multilevel domain name registration
- Synchronized sampling across all entities

ASAP Hybrid Discrete Object Thresholds (DOTs) Features

- Objective thresholding based on domain names and levels of the domain name
- Data sent to and objectives applied on a NonStop™ server
- Thresholds based on the DOTs operators >, <, =, <>, <=, and >= on any application metric
- Objectives thresholds and alerts reported by the ASAP Client based on the actual threshold
- User-configurable EMS threshold events

Data Collection and Monitoring

ASAP monitors and reports high-level, real-time network availability and user-defined application availability from one or more nodes. Availability information is sent to one or more collection nodes within the network. You use ASAP to view and analyze real-time or historical network-wide views of Application, CIP, Comm, CPU, Disk, Expand, File, Hybrid, Node, Process, ProcessBusy, RDF, Spooler, System, Tape, TCP/IP, Telserv, and TMF availability. You can customize each node for monitoring all or a subset of entity availability.

ASAP availability information is stored in the ASAP database for historic reference, as well as for shared access by multiple users and applications. The ASAP Server collects data with a set of CPU Monitors, Statistic Gathering processes (SGPs), and Collection processes in a network of systems. ASAP monitors operational information of system and user-defined resources. The ASAP SGPs report on performance and operational information of system and user application resources. This information is collected at defined intervals and forwarded to a Collection process residing on the same or another node in the network. The Collection process maintains this information in the ASAP database.

You can configure ASAP to gather performance and state information from multiple nodes throughout the network. You can independently configure these nodes so some nodes monitor Disk availability while other nodes do not. You can centralize data collection to a single node, to groups of regional nodes, or completely distribute it so each node in a network has its own ASAP availability database.

ASAP lets you monitor individual objects, by defining such objects with the MONITOR command.

For example, in a 10 processor node, all ASAP SGPs let you get availability information from all 10 CPUs or just from a few of them. You can separately set individual thresholds for each object, such as CPU 0 can report EMS events to alert an operator when the CPU is 45% busy, and send an alert if CPU 0 is less than 25% busy. Meanwhile, you can set other individual thresholds for CPU 1 and all remaining CPUs on the node.

ASAP also maintains time synchronization of nodes. You can configure it to synchronize time daily at a specific interval. Remote node times are synchronized with the Collection node.

The Collection process automatically creates and maintains the ASAP database files. You can configure database maintenance options to automatically release files, purge data, maintain a moving window of data over time, or turn off collection for a selected period of time.

The ASAP database is a real-time structured database, organized so that the most current network availability information is immediately available. If you need direct programmatic access to the ASAP database, see the record description of each ASAP entity in Appendix B, [Appendix B: ASAP](#) Data Definitions.

Availability Overview

The ASAP Server collects state and performance information for System and Application entities. ASAP entities that report on performance and state information include:

- Application domain statistics (through the optional ASAPX and ASAPH products) track the productivity, performance, and availability of customer-developed

applications. Service-level metrics are automatically evaluated against user-defined objectives to establish alert priorities when application objectives are not being met for an application process.

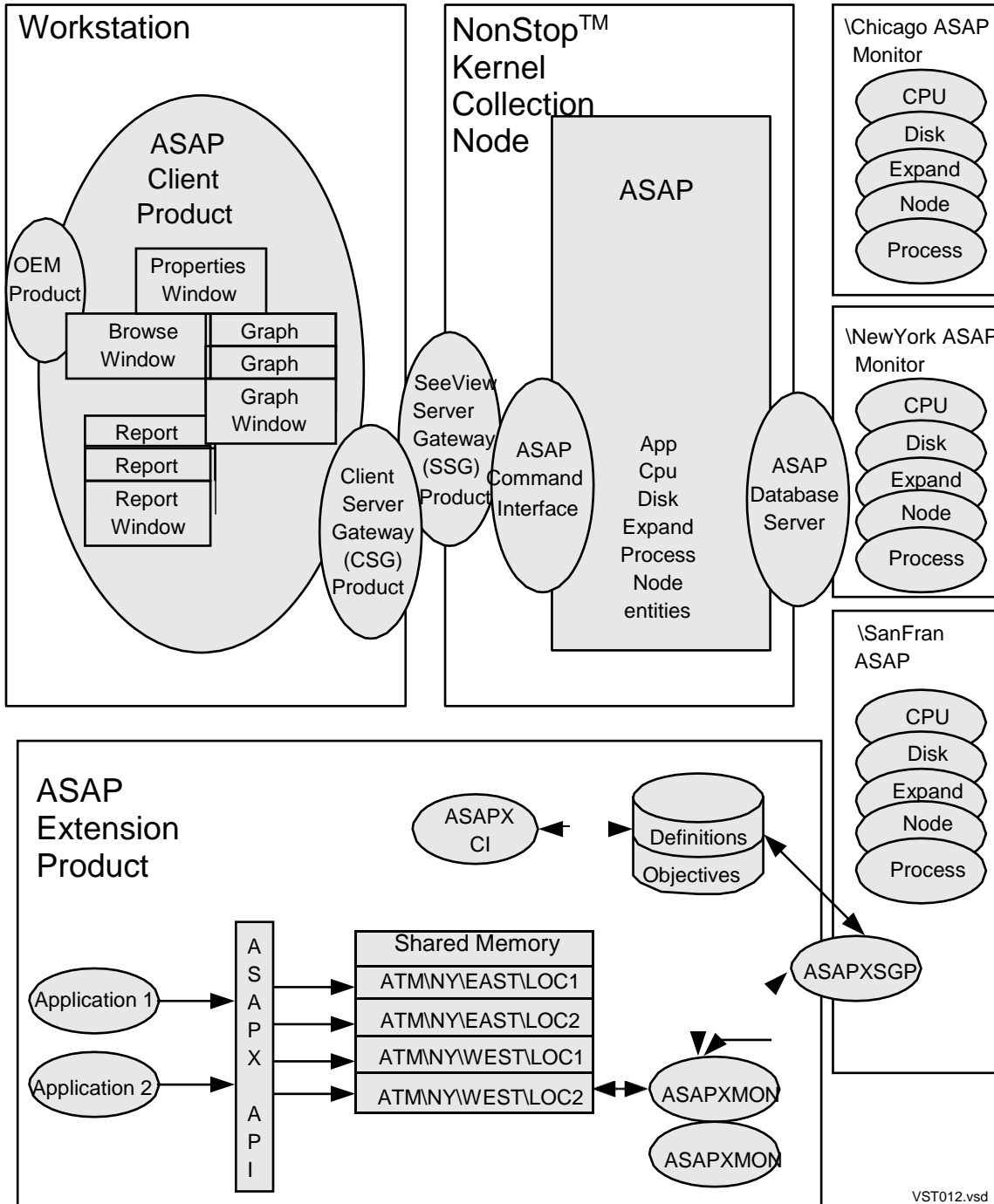
- CIP subsystem status and statistics for NonStop™ servers that use CLIM devices, including information for CLIM devices, Monitor and Provider processes.
- Communication line availability for AM3270, ATP6100, Envoy, EnvoyACP, Multilan, Snax, and X.25 such as percent line busy, Input/Output bytes read and written InputData/OutputData bytes read and received, and Retries, state of the line are reported by the COMM SGP.
- CPU availability, such as percent CPU busy, interrupt busy rates, send busy rates, queue lengths, dispatches, disk I/O, cache hits, and memory swap rates. State reporting of CPU up or CPU down is also supplied.
- Disk availability, such as percent disk busy, percent write busy, percent read busy, number of disk requests, capacity used, cache hits, queue lengths, and swap rates are a few of the performance statistics reported by the Disk SGP. State reporting about down paths to a Disk device, such as the Disk process, is also provided.
- Expand line-handler availability information, such as packets sent, received, and passed through; error rate statistics such as block check character errors (BCC), no buffer failures, and negative acknowledgment rates. State reporting of whether the Expand line-handler is up, down, or connecting is also provided.
- Expand end-to-end node availability, such as packets sent/received statistics and path information for number of hops from one Expand node to another.
- File availability information including existence, security settings, size, and amount of space remaining.
- Process availability information including location, priority, and resource consumption.
- ProcessBusy availability information, such as busiest processes, CPU busy percent per process, messages sent and received, receive queue length, and average memory pages used, are a few of the performance statistics reported by the ASAP Monitor.
- User selected Files collects file availability information, such as percent full, EOF, file code, and file format.
- User selected Processes collects process availability information, such as CPU, PIN Pri, Busy, Process state, Wait State, and Pages used.

- RDF statistics, such as relative delay times, relative byte address of the record being processed, and sequence number of the audit or image file.
- Spooler availability information, such as jobs, open, hold, and print.
- Swap availability information from the KMSF subsystem such as total swap space, allocates, creates, frees, releases, resizes, used percentage, number of files and available pages.
- System availability information, such as percent system busy, queue lengths, and memory swap rates.
- Tape availability information, such as mounts, tape status, and tape label.
- TCP/IP availability and statistics information for the ARP, ICMP, ICMP6, IGMP, IP, IP6, Loop, MonGQ, Port, Process, QIO, Route, RTE, Socket migration, Subnet and UDP layers of the TCP/IP subsystem.
- Telserv availability and statistics monitoring of processes, services and windows, including usage and connection information.
- TMF statistics, such as transactions per second, percentage used of the audit trail, and catalog status.
- All ASAP SGPs can report discrete objective thresholds for most of the SGP statistics using the operators >, <, =, <>, <= and >=. You can set thresholds to be reported as EMS events when they do not meet the user-defined objective.

Components of ASAP

ASAP has several major functional components, as the following figure shows. For more information about the ASAP Client, see the ASAP Client Manual. For more information about the OEM, navigate to Program Files\Tandem\OEM on workstations that have the ASAP Client installed.

Figure 1-1. ASAP Components



Presentation Interfaces

The basic ASAP product includes the ASAP Client and Server, shipped on two CDs.

Conversational Interface

The ASAP Conversational Interface (CI) provides a command interpreter so you can work directly from the collection system. The CI also provides command and control of the ASAP components. See [Section 6: ASAP Conversational Interface Commands](#).

The ASAP CI provides interactive, formatted reports for all monitored nodes or selected nodes. Reports are obtained by entering commands such as APP, COMM SGP, CPU, Disk, File, LH(Expand), Process, Processbusy, RDF, Spooler, System (cpu, avg), Tape, or TMF. These commands generate reports that display the specified entity.

You can use the ASAP Conversational Interface to get status information about the monitor process, the collector process, and the statistic gathering processes. The ASAP CI is also used for configuring, starting, and stopping the ASAP environment.

Graphical User Interface

The ASAP Client provides a graphical user interface (GUI) for in-depth analysis of ASAP availability information. It displays color-coded Graph, Grid, and Tree Views of monitored nodes, entities, and objects, and its easy-to-use interface helps you understand the status and performance of your systems and applications.

The ASAP Client displays both real-time and historical views of Application, CPU, Disk, Expand, File, Node, Process, ProcessBusy, RDF, Spooler, System, Tape, and TMF entities. When objects are down, they are annotated with color-coded icons. These icons are propagated upward through all visual tree hierarchies, so you are always aware of an object's state, no matter what level of the tree is visible.

You can use the Object State Change (OSC) window to obtain a history of object state changes. The OSC view provides a high-level graphical overview of object state changes and a detailed log of state changes. The OSC window is divided into two areas so you can get both high-level and detailed state change information.

When the ASAP Extension (ASAPX) option is present, the ASAP Client also displays application statistics and domain service-level metrics. Informational views can encompass all nodes, a selected set of nodes, or focus on a single node, entity, or object.

Batch Reporting

The Enform product provides ad hoc batch reporting and analysis of the central ASAP database. You can use the collect process as an Enform server to get fast hourly rollup reporting.

ASAP performance information is stored in a structured real-time availability database that user programs can easily access. The structure of this normalized database is documented in each release of ASAP. To generate DDL data structures that you can

use in your program for accessing the ASAP database, refer to the released file ASPDDLDB.

Command and Control

The ASAP Conversational Interface provides commands and options that let you work interactively with the collection system:

- STATUS, STARTUP, and SHUTDOWN commands control the ASAP Monitor and Collector processes. When any entity is configured for gathering entity availability information, the ASAP Monitor process also starts the selected entities.
- The SET command controls the ASAP environment and is also used for setting individual SGP options. See the [SET Command](#).
- Display formats for application entities are defined in EDL.
- All entity commands display current availability reports or reports from different times when the command is followed by TIME *hh:mm*.

Database

The ASAP database contains current and historic normalized system, Application, COMM SGP, CPU, Disk, Expand line-handler, File, Node end-to-end, Process, ProcessBusy, RDF, Spooler, Tape, and TMF availability information. The database records for all entities also contain operational state information.

Database features include:

- Quick retrieval of statistical information.
- Automatic long-term availability and performance history data.
- No remote disk space requirement for availability information. Remote object file space is required.
- Database maintenance functions:
 - Fast Cleanup for immediate deletion of all records.
 - Daily deletion of all records at a user-specified time.
 - Daily retention of recent records. You define the number of records to retain.

- The Rollover option renames entity database files daily so you can archive them for historic analysis at a later date.

Once started, ASAP needs no operator intervention. Even when CPUs become unavailable, the ASAP database is automatically maintained.

Collection

The Database Collector is a process pair that collects application and system availability information from system nodes throughout the network. Availability information records from remote monitors and SGPs throughout the network are received by the ASAP Collector process, which stores the data in the normalized ASAP database.

User-defined partitions let multiple volumes share the I/O workload.

Node Monitors

Each monitored node in the network has a Monitor process pair running in it. The Monitor starts individual SGPs based on user criteria for each SGP. If the SGP CPU becomes unavailable after starting, the monitor restarts the SGP in an available CPU.

Statistics Gathering Processes

Each CPU on the monitored node has at least one Statistics Gathering Process (SGP) executing in it provided that monitoring is turned on, or auto discovery was used. This SGP process determines CPU availability and busiest Process information and then sends these availability records to the Collector process. When at least one CPU on the node is monitored, system availability information is collected. System availability is the average availability statistics for all monitored CPUs on the node.

Application Statistics Gathering Process

The application SGP collects customer application service-level metrics from application monitor processes for tracking productivity, performance, and availability of the applications. These service-level metrics are automatically evaluated against predefined objectives for establishing alert priorities, letting system operators and administrators quickly identify service-level issues that are not meeting the defined objectives for an application process. This SGP, the ASAPX Monitor processes, the ASAPX CI, and the ASAPX Library are all part of the optional product ASAPX.

CIP Statistics Gathering Process

CIP SGPs collect availability information on Cluster I/O Module (CLIM) devices on NonStop™ servers. The CIP SGP collects status and statistics information for CLIM devices, CIP Monitor processes and CIP Provider processes. It uses several

different sub-entities and ASAP record types to report CIP information to the ASAP Collector process.

COMM Statistics Gathering Process

Communication SGPs collect availability for all communication lines configured on the node or only user-specified communication lines on the node. This process uses the Measure subsystem programmatic interface for collecting raw communication performance counters. The COMM SGP reports on objective thresholds that exceed the user-defined values maintained in the objectives database.

The COMM SGP gathers availability information, bundles it in a single message, and forwards it to the ASAP Collector process. The Collector process unpacks the information and stores it in the normalized ASAP database so the ASAP COMM command or ASAP client can retrieve it.

CPU Statistics Gathering Process

The remote ASAP Monitor starts the CPU SGP in each CPU of the monitored node. CPU SGPs are responsible for gathering CPU-level statistics, including CPU performance metrics, Operational state, and information about the busiest processes in their CPU.

The CPU SGPs reports CPU-level information to the ASAP Monitor process. The Monitor bundles it in a single message and forwards it to the ASAP Collect process on the main Collection node. The Collect process parses this information and stores it in the normalized ASAP database. You can then display the data with the ASAP CPU and PROCESSBUSY commands, or with various block-mode or workstation programs.

Disk Statistics Gathering Process

Disk SGPs collect disk availability information for all non-phantom disks on the node. These processes use the Measure subsystem programmatic interface for collecting raw disk process counters. They do not use the Measure subsystem for determining performance counters, disk capacity utilization, or operational state information. The Disk SGP reports on objective thresholds when they do not meet the user-defined values.

The Disk SGP gathers availability information, bundles it in a single message, and forwards it to the ASAP Collect process. The Collect process unpacks the information and stores it in the normalized ASAP database so the ASAP DISK command or ASAP Client can retrieve it.

Expand Statistics Gathering Process

Expand SGPs collect Expand availability information for all Expand lines on the node. These processes use the Subsystem Programmatic Interface (SPI) for communicating

with the Expand Manager. The Expand SGP reports on objective thresholds when they do not meet the user-defined values.

The Expand SGP gathers availability information, bundles it in a single message, and forwards it to the ASAP Collect process. The Collect process unpacks the information and stores it in the normalized ASAP database so the ASAP LH command or ASAP Client can retrieve it.

File Statistics Gathering Process

File SGPs collect file availability information. You define which files or subvolumes the File SGP monitors. File availability information is obtained using the Guardian API. The File SGP reports on objective thresholds when they do not meet the user-defined values.

File monitoring in ASAP 2.5 has been enhanced to provide logical grouping of files, aggregation of file domains, and the ability to restrict the historical database to contain only aggregate records for certain file groups.

Node Statistics Gathering Process

Node SGPs collect Expand end-to-end node statistics for the Node entity. These processes use the Subsystem Programmatic Interface (SPI) to collect path information and traffic statistics between two Expand end nodes.

The Node SGP gathers availability information, bundles it in a single message, and forwards it to the ASAP Collect process. The Collect process unpacks the information and stores it in the normalized ASAP database so the ASAP NODE command or ASAP Client can retrieve it.

Process Statistics Gathering Process

Process SGPs collect availability information about user selected processes. Process availability information is obtained using the Guardian API. The Process SGP reports on objective thresholds when they do not meet the user-defined values.

Process monitoring in ASAP 2.4 has been enhanced to provide logical grouping of processes, aggregation of process domains, and the ability to restrict the historical database to contain only aggregate records for certain process groups.

RDF Statistics Gathering Process

RDF SGPs collect RDF availability information about the Extractor and Duplicator nodes. The RDF SGP reports on objective thresholds when they do not meet the user-defined values.

Spooler Statistics Gathering Process

Spooler SGPs collect Spooler availability information. Process availability information is obtained using the Spooler API. The availability information gathered is performance utilization and operational state information. The Spooler SGP reports on objective thresholds when they do not meet the user-defined values.

The Spooler SGP gathers availability information, bundles it in a single message, and forwards it to the ASAP Collector process. The Collector process unpacks the information and stores it in the normalized ASAP database so the ASAP SPOOLER command or ASAP Client can retrieve it.

SWAP Statistics Gathering Process

Swap SGPs collect KMSF availability information. Swap availability information is obtained from the NSKCOM interface using the Seeview Server Gateway. The availability information gathered is performance utilization and operational state. The Swap SGP reports on objective thresholds when they do not meet the user-defined values.

The Swap SGP gathers availability information, bundles it in a single message, and forwards it to the ASAP Collect process. The Collect process unpacks the information and stores it in the normalized ASAP database so the ASAP SWAP command or ASAP Client can retrieve it.

Tape Statistics Gathering Process

Tape SGPs collect Tape availability information. Tape availability information is obtained from the Media Serve process using the Subsystem Programmatic Interface (SPI). The availability information gathered is performance utilization and operational state. The Tape SGP reports on objective thresholds when they do not meet the user-defined values.

The Tape SGP gathers availability information, bundles it in a single message, and forwards it to the ASAP Collect process. The Collect process unpacks the information and stores it in the normalized ASAP database so the ASAP TAPE command or ASAP Client can retrieve it.

TCP/IP Statistics Gathering Process

TCP SGPs collect availability information on layers of the TCP/IP stack, on Subnets and Routes, and on Telserv processes, services and windows. The TCP SGP supports sixteen different TCP/IP entities representing different layers and functions of TCP/IP and also supports three entities representing the Telserv subsystem.

Hundreds of statistics are gathered, analyzed, reported on and alerted on for the various TCP/IP and Telserv entities.

TMF Statistics Gathering Process

TMF SGPs collect TMF availability information. TMF availability information is obtained using the Subsystem Programmatic Interface (SPI) for the TMFSERVE process. The availability information gathered is performance utilization and operational state. The TMF SGP reports on objective thresholds when they do not meet the user-defined values.

The TMF SGP gathers availability information, bundles it in a single message, and forwards it to the ASAP Collect process. The Collect process unpacks the information and stores it in the normalized ASAP database so the ASAP TMF command or ASAP Client can retrieve it.

CSG

The Client Server Gateway works with the SeeView Server Gateway to provide a bridge between programs running on Windows NT, Windows 95, or Windows98 systems and programs running on NonStop™ OS systems. Through the CSG, client applications have access to NonStop™ OS server programs (such as the SSG) and to the HPE NonStop™ Advanced Command Language (TACL).

SSG

The SeeView Server Gateway provides a Client-Server Command Interpreter gateway. The SeeView Server Gateway is a SeeView ServerMode script that controls a SeeView process running in ServerMode.

COLLECT/ASAPDB

The ASAP database includes current and historical information about system performance and the operational status of system objects. The ASAP database is automatically created when the ASAP Collect process is first started. The Collect process receives statistics and status messages from the ASAP and ASAPX monitors throughout the network, parses the messages, and stores the information in the normalized ASAP database.

The Collect process automatically maintains the database. The Collect process includes database maintenance options that specify how much history to retain. You can configure database history to only retain data since the last cleanup time; for example, every night at midnight. Or you can configure it to retain ASAP history for a much longer period of time, even indefinitely.

For more information about database maintenance options, see the SET RETAIN command, SET DBMAXDAYS, SET DBMaxDaysToPurge,

SET DBRolloverSubvolPrefix, and the SET DBRolloverFilePrefix commands beginning with the [SET Command](#) on page [251](#).

The ASAP Collector process is continuous. If a CPU fails, the backup Collector process automatically takes over so that database access is always available. The ASAP database is a real-time normalized database and can be queried online. The ASAP product subvolume includes Enform queries that allow batch-type queries of network performance and operational status.

Section 2: Installing ASAP on the Server

Topic	Page
Task 1: Ensure Your System Meets Requirements	46
Task 2: Install ASAP Server Product Files	47
Task 3: Set Up ASAP Server	48
Task 4: Configure ASAP Components	58
Starting and Stopping ASAP	71

Task 1: Ensure Your System Meets Requirements

To install ASAP Server, your system must meet these requirements:

Hardware

HPE NonStop™ server. For more information, see the ReadMe file included with your RVU or software product revision (SPR) for ASAP.

Terminals and Terminal Emulators

- HPE TS530G12 terminal or any earlier terminal type for the NonStop™ server
- Crystal Point Outside View terminal emulator version 4.2 or later
- TTE PC version or later
- PC6530 PC emulator B40 or later
- MacMenlo Macintosh emulator 4.0 or later

Software

- NonStop™ OS T9050D30 or later
- DDL T9100, any D-series RVU or later

- Expand Manager, T9117D30 version AAW or later
- Expand line-handler, T9057D30 version ADE or later
- HPE NonStop™ SeeView, T6965C31 or later
- DSM/Tape Catalog MediaSRV server, T6024D42 version AAT or later

Task 2: Install ASAP Server Product Files

To install ASAP product files on the system, use the IPSETUP program on the ASAP installation CD. For more information on IPSetup, see the IPSetup help on the CD.

1. Put the ASAP installation CD in your workstation.
 - If AutoRun is enabled on your workstation, IPSETUP starts automatically after you insert the CD.
 - If AutoRun is not enabled, display the contents of the CD and double-click IPSETUP.exe.

IPSetup displays its welcome screen.

2. Click **Continue**.

IPSetup displays the Software License Agreement.

3. Read the License Agreement; then click **Accept** to accept the stated usage conditions.

IPSetup prompts you to select your NonStop™ Kernel processor type.

4. Select NonStop™ Kernel RISC or NonStop™ Kernel CISC. IPSetup displays the Available Product list.

5. Select ASAP Server in the list; then click **Add** to move ASAP Server to the Selected Products list.

6. Enter the host system name, your user name, and password.

7. Enter `$volume.subvolume`.

IPSetup displays the Destination Location dialog box.

8. To use the displayed location, click **Next**. To specify a different location, click **Browse**.

IPSetup displays the Destination Browser.

9. Choose the location and click **OK**.

IPSetup returns you to the Destination Location dialog box.

10. Click **Next**.

IPSetup transfers the files to the designated volume and subvolume and prepares them to be used as DSVs. IPSETUP displays status dialogs so you can monitor the progress of the host installation.

When IPSetup is finished, it prompts you to indicate how you want to conclude your IPSetup session.

11. Indicate how you want to conclude your IPSetup session:

- To view release documentation when IPSetup closes, click the **View Release Documentation** checkbox.
- The **Launch DSM/SCM Planner Interface** check box is checked if you indicated before that you will manage the product with DSM/SCM. If you do not want to launch DSM/SCM when IPSetup closes, uncheck the check box.

12. Click **Finish**.

IPSetup closes and, as specified, opens the review documentation if the **View Release Documentation** check box was checked, or launches DSM/SCM.

Task 3: Set Up ASAP Server

The ASAP installation subvolume `$volume.ZASAP` that was created from IPSetup includes an OBEY file named `INSTALL`, which invokes the `ASPSETUP` script.

`ASPSETUP` is a SeeView script that automates the setup and configuration of all required ASAP components.

1. Verify that SeeView D30 or later is installed on your system.

To install SeeView on your system, use the `Install` product or use the `OBEY` file named `INSTALL` on the SeeView distribution subvolume.

2. Use the `ASPSETUP` script to install ASAP on all nodes to ensure proper file locations and security settings:

- To run `ASPSETUP` as a SeeView external, see [Section 2: Installing ASAP on the Server](#), in the *SeeView Server Gateway (SSG) Manual*.
- To run `ASPSETUP` from a TACL prompt:

```
TACL 1> VOLUME $volume.subvolume.ZASAP
```

```
TACL 2> OBEY INSTALL
```

`ASPSETUP` displays the **AsapInstall** menu (see the following figure).

To invoke any menu options, move the cursor to the menu item and press Return.

For context-sensitive help for any menu item or field in ASPSETUP, move the cursor to the item or field and press Shift-Return.

3. Choose the type of installation to perform:

- To fully install and configure ASAP on a data collection node, proceed to [Full Install on This Node](#).
- To install ASAP on remote nodes so you can monitor other nodes at the central data collection node, proceed to [Remote Monitor Install](#).

Figure 2-1. ASPSETUP Main Menu, AsapInstall

```
AsapInstall
■ Full Install on this node.
  Remote Monitor Install.

System subvol Install only.
Database subvol Install only.
ASAP Tacl window.
EXIT and Save Log.

AsapSetup (16OCT98) - Copyright Compaq Computer Corporation 1999

Select one of the following options from the AsapInstall menu above
by moving the cursor to the desired menu item and pressing <RETURN>.

-Full Install on this node.   Install files on this COLLECT node.
-Remote Monitor Install.     Install files on remote MONITOR nodes.

-System subvol Install.      Partial install $System.System subvolume only.
-Database subvol Install.    Partial install Database subvolume only.
-EXIT and Save Log.          Exit setup and save log to ZMddHHMM.

N O T E !! You can obtain context sensitive help for any menu-item or
field by moving the cursor to that item and then pressing <SHIFT-RETURN>.
```

VST002.vsd

Full Install on This Node

The Node installation copies object files and configuration files to the \$SYSTEM.SYSTEM location. The install process secures the files for network access, and creates the database DDL dictionary.

To fully install and configure ASAP on a data collection node:

1. From the **AsapInstall** menu, select **Full Install on this node**; then press Return.

ASPSETUP displays the ASAP Local Install Configuration fields (see the following figure).

Figure 2-2. Installation Configuration for Full Install on This Node

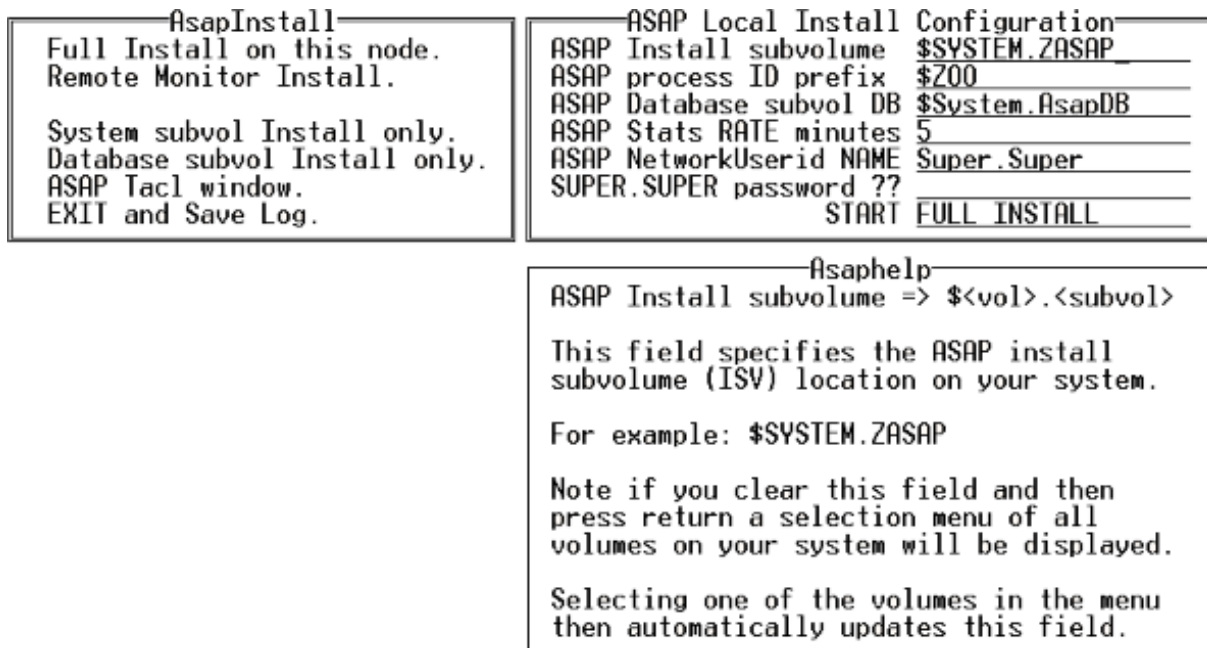
<pre>AsapInstall Full Install on this node. Remote Monitor Install. System subvol Install only. Database subvol Install only. ASAP Tacl window. EXIT and Save Log.</pre>	<pre>ASAP Local Install Configuration ASAP Install subvolume \$SYSTEM.ZASAP ASAP process ID prefix \$Z00 ASAP Database subvol DB ASAP Stats RATE minutes 5 ASAP Networkuserid NAME SUPER.SUPER password ?? START FULL INSTALL</pre>
<pre>Asaphelp Please fill out the fields in the Installation Configuration panel above. NOTE !! You can obtain context sensitive help for any field above by selecting that item with the cursor and then pressing SHIFT-RETURN.</pre>	

VST003.vsd

For context-sensitive help on any field in the ASAP Local Install Configuration panel, put the cursor on that field and press Shift-Return. The following figure shows the result of pressing Shift-Return with the cursor on the ASAP Install subvolume field.

ASPSETUP displays default values in uppercase when it displays the configuration panel. Values shown in upper and lowercase are user-specified values and are not present when ASPSETUP first displays the configuration panel.

Figure 2-3. Input Help for the ASAP Install subvolume Field



VST011.vsd

2. Enter the Local Install values for configuring ASAP on the local node.
To start a full installation:
 - a. Enter the ASAP installation location in the ASAP Install subvolume field.
 - b. Enter a process ID.
 - c. Enter the location you want for the ASAP database `$volume.subvolume`.
 - d. Enter the Statistics Collection Rate.
 - e. Enter a network user ID name.
 - f. Enter the password for SUPER.SUPER.
 - g. Press Return to start the installation.

The following figure shows a sample first page of output when you start a full installation.

Figure 2-4. First Page of Output From the START FULL INSTALL Function

```

AsapInstall
Full Install on this node.
Remote Monitor Install.

System subvol Install only.
Database subvol Install only.
ASAP Tacl window.
EXIT and Save Log.

ASAP Local Install Configuration
ASAP Install subvolume $SYSTEM.ZASAP
ASAP process ID prefix $ZOO
ASAP Database subvol DB $System.AsapDB
ASAP Stats RATE minutes 5
ASAP Networkuserid NAME Super.Super
SUPER.SUPER password ??
START FULL INSTALL

Asap Tacl
06:31:59 STEP #1 = Checking security
Logon name: SUPER.SUPER
$SYSTEM ZASAP 3> who
Userid: 255,255 Username: SUPER.SUPER Security: "UUUU"
06:32:00 STEP #2 = Checking $SYSTEM.ZASAP ISV subvol
06:32:01 STEP #3 = Checking $SYSTEM.ASAPDB Database subvol
Logon name: SUPER.SUPER
$SYSTEM ZASAP 4>
06:32:02 STEP #4 = Checking ASAPCONF file/s
06:32:02 Creating $SYSTEM.SYSTEM.ASAPCONF
06:32:03 Creating $SYSTEM.ASAPDB.ASAPCONF
06:32:04 STEP #5 = Building ASAP installation obey file
06:32:04 Building System install commands.

```

VST004.vsd

ASPSETUP displays an **Asap Tacl** window so you can monitor the progress of the installation on the collection node. During the installation, ASPSETUP:

- Performs security checks.
- Checks for a previous version of ASAP. If a previous version is found, the older version is renamed to \$*volume*.ZASAPOLD.*.
- Checks for the existence of an ASAP database.
- Creates Configuration files, including an installation OBEY file.
- Shuts down the existing ASAP system if it is running.
- Starts the installation process.

The installation terminates if you exit SeeView while ASAP is being installed. To terminate the installation without exiting SeeView, from the **AsapInstall** menu, select **Abort Install**, and press Return.

When the Full Install completes, ASPSETUP displays output similar to the following figure.

Figure 2-5. Completion Display From the START FULL INSTALL Function

```

AsapInstall
Full Install on this node.
Remote Monitor Install.

System subvol Install only.
Database subvol Install only.
ASAP Tacl window.
EXIT and Save Log.

ASAP Local Install Configuration
ASAP Install subvolume $SYSTEM.ZASAP
ASAP process ID prefix $Z00
ASAP Database subvol DB $System.AsapDB
ASAP Stats RATE minutes 5
ASAP Networkuserid NAME Super.Super
SUPER.SUPER password ??
START FULL INSTALL

Asap Tacl
DICTOUK      208      12288 05JAN2000  7:01 255,255 NUNU      4      32
DICTRDF      205      12288 05JAN2000  7:01 255,255 NUNU      4      32
DICTTKN      209           0 05JAN2000  6:59 255,255 NUNU      4      32
DICTTYP      209           0 05JAN2000  6:59 255,255 NUNU      4      32
DICTVER      209           0 05JAN2000  6:59 255,255 NUNU      4      32

comment check output for errors.
comment ASAPOBEY install complete.
$SYSTEM ASAPDB 11>

```

VST005.vsd

3. To confirm that the installation did not encounter any errors, scroll through the output in the **Asap Tacl** window.

The **Asap Tacl** window is a SeeView window that scrolls automatically. To turn off the automatic scroll feature, press Delete-Line from a terminal or Ctrl-Delete from a workstation.

Remote Monitor Install

A Remote Monitor Install copies the appropriate files to one or more remote nodes, secures the object for network access, and sets the program ID to the network user ID.

Note

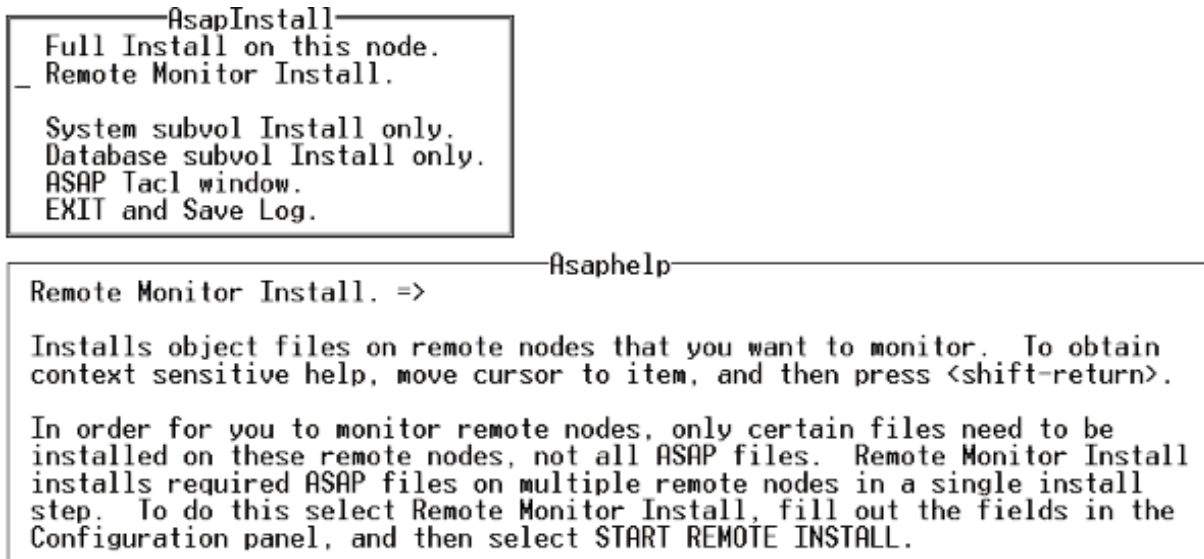
To use context-sensitive help, put the cursor in the Remote Monitor Install field and press Shift-Return. The following figure displays the ASAP context-sensitive help text for the **Remote Monitor Install** menu item.

To install ASAP on remote nodes so you can monitor other nodes at the central data collection node:

1. From the **AsapInstall** menu, select **Remote Monitor Install**; then press Return.

ASPSETUP displays the **Asap Remote Install Configuration** fields (see the following figure).

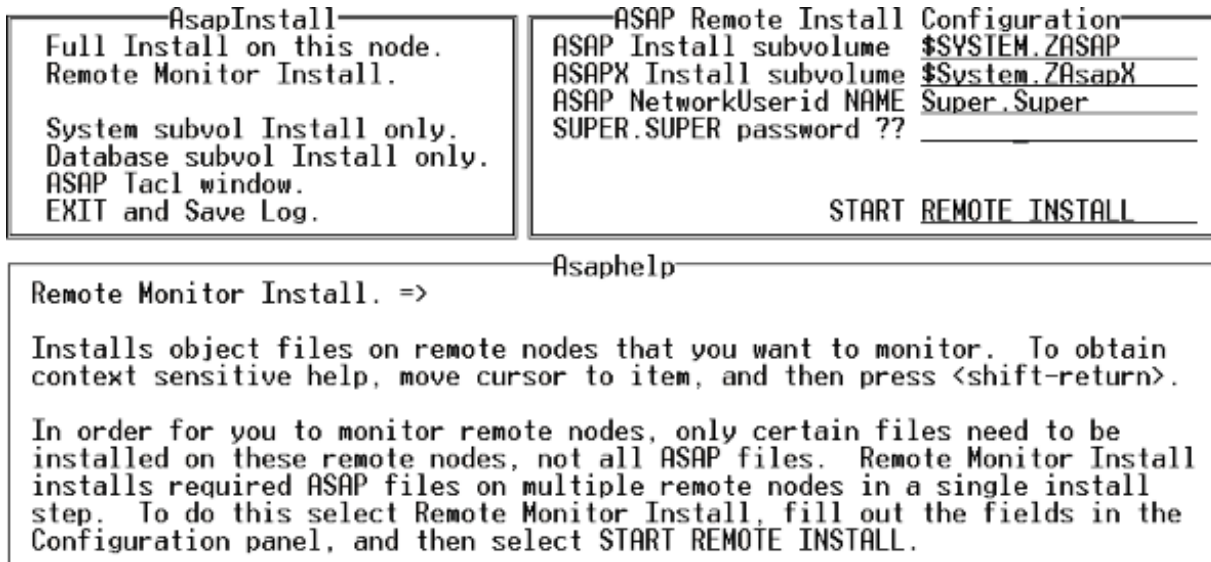
Figure 2-6. ASAP Help Text for Remote Monitor Install



VST006.vsd

The following figure shows a sample **Remote Monitor Install** page. Uppercase characters represent default values. Mixed case character entries are supplied by the user and have no default values.

Figure 2-7. Remote Monitor Install Display With User Entries



VST007.vsd

You do not have to install both ASAP and the ASAP Extension (ASAPX) on remote nodes. To install one but not the other, leave the **ASAP Install subvolume** or **ASAPX Install subvolume** field blank for the product that you do not want to install.

2. Enter the ASAP installation location in the **ASAP Install subvolume** field.
3. Optionally, enter the ASAP Extension subvolume in the **ASAPX Install subvolume** field.
4. Enter a network user ID.
5. Enter the password for SUPER.SUPER.
6. Press Return to start the installation.

ASPSETUP displays a menu of all the nodes in your network (see the following figure).

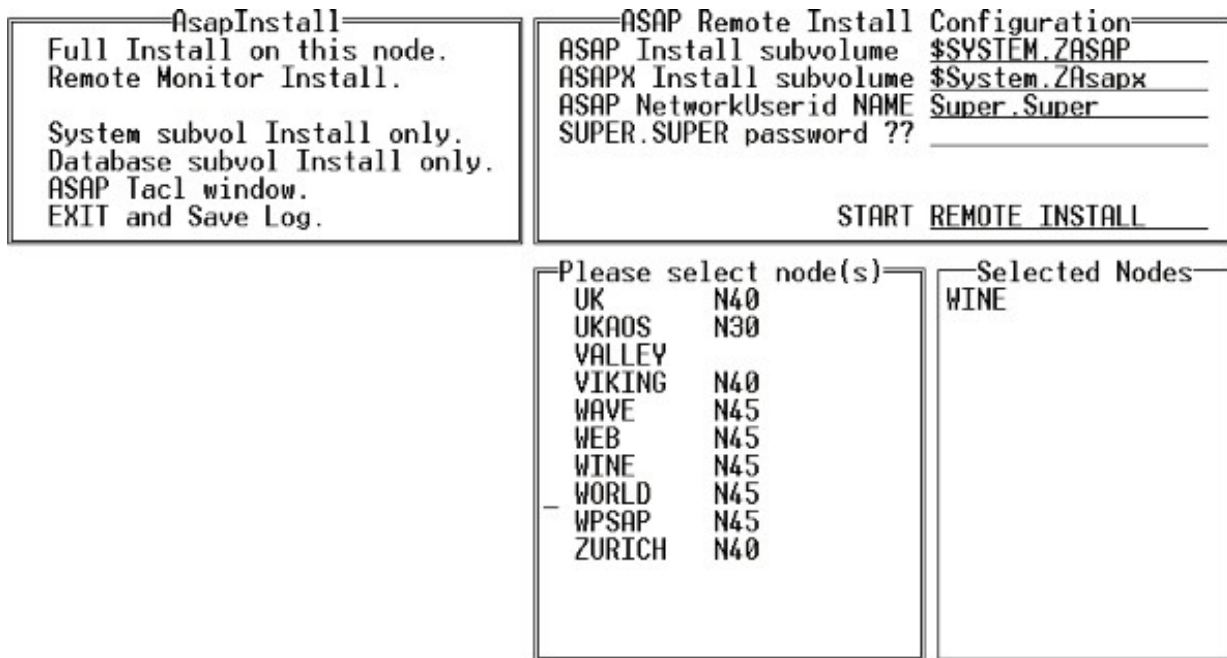
7. To specify the nodes you want to monitor remotely with ASAP, move the cursor to the node name; then press Return.

When you select a node, its name is placed in the **Selected Nodes** window. You can select any number of nodes.

To deselect a node, move the cursor to the node in the **Selected Nodes** window and then press Return. The node name is removed from the **Selected Nodes** window.

The following figure shows the results of installing ASAP and the ASAP Extension on the node WINE.

Figure 2-8. Selecting Remote Install Nodes for Remote Installation



VST008.vsd

- After you select all of the nodes to install remotely, move the cursor back to the **START REMOTE INSTALL** field and press Return.

ASPSETUP shuts down the remote monitors on the selected nodes, creates an installation OBEY file, and starts the remote installation process (see the following figure).

Figure 2-9. Output From Remote Install

```

AsapInstall
Full Install on this node.
Remote Monitor Install.

System subvol Install only.
Database subvol Install only.
ASAP Tacl window.
EXIT and Save Log.

ASAP Remote Install Configuration
ASAP Install subvolume $SYSTEM.ZASAP
ASAPX Install subvolume $System.ZAsapx
ASAP NetworkUserid NAME Super.Super
SUPER.SUPER password ??

START REMOTE INSTALL

Asap Tacl
08:54:34 STEP #1 = Checking security
Logon name: SUPER.SUPER
$SYSTEM.ZASAP 4> who
Userid: 255,255 Username: SUPER.SUPER Security: "NUUU"
08:54:35 STEP #2 = Checking $SYSTEM.ZASAP ISV subvol
08:54:35 ASAP installation subvol appears OK
08:54:35 STEP #3 = Checking $SYSTEM.ZASAPX ISV subvol
08:54:35 ASAPX Install Subvol appears OK
Logon name: SUPER.SUPER
08:54:35 STEP #4 = Building ASAP installation obey file
$SYSTEM.ZASAP 5>
08:54:35 STEP #5 = Saving installation commands in file AspNodes
08:54:36 STEP #6 = Shutdown old ASAPMon on remote nodes

```

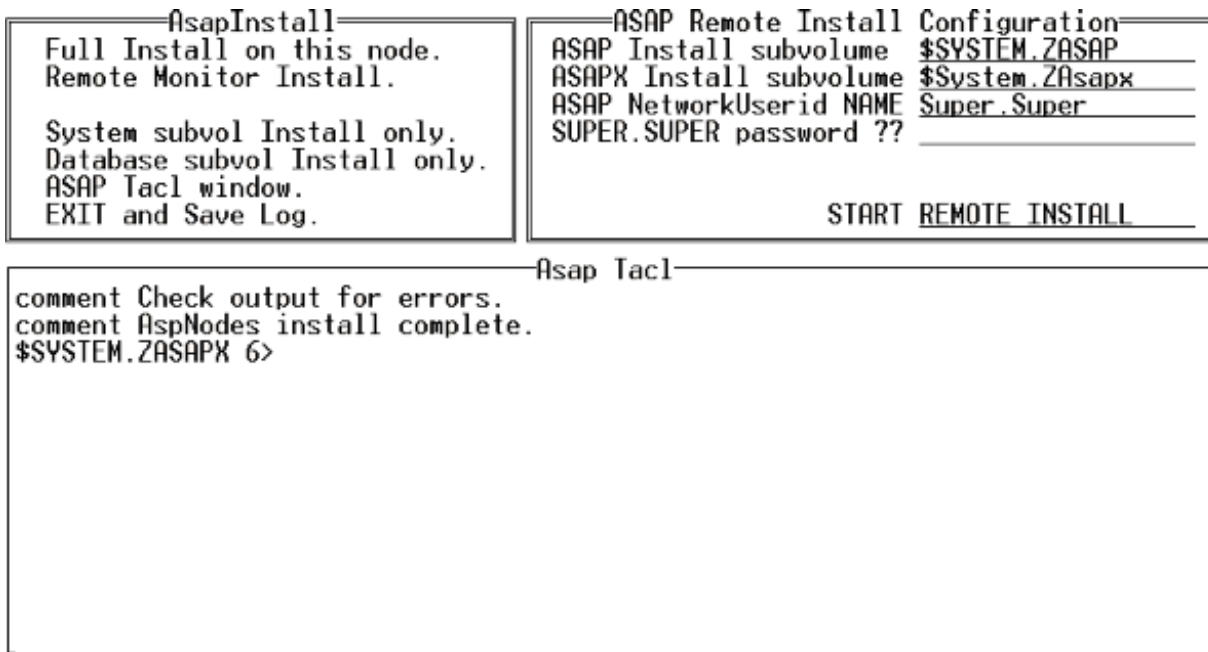
VST009.vsd

ASPSETUP displays an **Asap Tacl** window so you can monitor the progress of the installation on each node that you selected.

The installation terminates if you exit SeeView while ASAP is being installed. To terminate the installation without exiting SeeView, from the **AsapInstall** menu, select **Abort Install**, and press Return.

When the install completes, ASPSETUP displays output similar to the following figure.

Figure 2-10. Sample Remote Install Completion Screen



VST010.vsd

9. To confirm that the installation did not encounter any errors, scroll through the output in the **Asap Tacl** window.

The **Asap Tacl** window is a SeeView window that scrolls automatically. To turn off the automatic scroll feature, press Delete-Line from a terminal or Ctrl-Delete from a workstation.

Task 4: Configure ASAP Components

To customize the ASAP configuration on your nodes, use this information about component configuration:

- ASAP configuration file
- ASAP object files
- ASAP database

Confirm ASAP Configuration Settings

ASAPCONF, the ASAP configuration file, sets up configuration information specific to your and users' sites. When the ASAP Conversational Interface (CI) starts, it searches for an ASAPCONF file and implements its instructions.

Any system with an active ASAP database should have a properly configured ASAPCONF file in the \$SYSTEM.SYSTEM location, in addition to any user configuration files located on other subvolumes. When the CI cannot locate the ASAPCONF file, ASAP uses the default values you define with the SET command (see the [SET Command](#)).

ASAP searches the current default subvolume, then \$SYSTEM.SYSTEM for the configuration file. If ASAP does not find the ASAPCONF file, the four most important values to specify from the ASAP CI and to use when creating the ASAPCONF file are:

- The SET DB command for defining a location for ASAP database files
- The VOLUME command for defining the location of the ASAP object file locations
- The SET RATE command for defining the availability sample rate interval
- The SET RETAIN command for defining the database maintenance option

This example creates a new configuration file named \$SYSTEM.SYSTEM.ASAPCONF. If you used the ASPSETUP Full Install on this node, the ASAPCONF can be found on \$SYSTEM.SYSTEM.

```
1 >COMMENT Create and setup the ASAPCONF file.
```

```
2 >EDIT $SYSTEM.SYSTEM.ASAPCONF; ADD
```

```
CURRENT FILE IS $SYSTEM.SYSTEM.ASAPCONF
```

```
1 SET DB $DATA.ASAPDB.DB !Set database name
```

```
2 VOLUME $DATA.ASAPDB !Set ASAP subvolume
```

```
3 SET RATE 5 !Set sample interval
```

```
4 SET RETAIN NONE !Set cleanup option
```

```
3 >FUP SECURE $SYSTEM>SYSTEM.ASAPCONF, "Nxxx"
```

After confirming the settings in the ASAPCONF file, you can run ASAP on your system.

You can use the ASAP configuration wizard, ASAPWIZ, as an alternative to manually creating an ASAPCONF file. To use the configuration wizard, change your volume to the ASAP installation subvolume and type RUN ASAPWIZ. The wizard determines the latest copy of ASAP installed on your system and prompts you with specific questions to properly configure ASAP in your environment.

Example ASAPWIZ session:

```
1> RUN ASAPWIZ
ASAP Configuration Wizard - T0403V02.05 - (15OCT2004) - System
\CENTDIV
Copyright (C) 1999, 2001-2004 Hewlett-Packard Enterprise

This wizard will assist you in creating an ASAP configuration
file (ASAPCONF). It should be run after installing ASAP.

Is this node, \CENTDIV, the central ASAP collection node(Y):y
ASAP 2.05 is installed in $SYSTEM.SYSTEM

Do you want to build the ASAP configuration for that ASAP
version(Y)?y

Please enter a subvolume for the ASAP
database($SYSTEM.ZASAP):$asap.testdb

*** There are files in the subvolume you specified

$ASAP.TESTDB

          CODE      EOF  LAST MODIFIED  OWNER  RWE  PExt  SExt
ASAPCONF  101      2048 11MAR2004   8:42 255,34 NUUU    8   32

Are you sure you want to use this subvolume(N):n

Please enter a subvolume for the ASAP
database($SYSTEM.ZASAP):$asap.testdb2

Please enter the primary CPU for the ASAP Monitor(1):
Please enter the backup CPU for the ASAP Monitor(0):

Would you like other ASAP processes to run in the same CPU as
the ASAP Monitor(Y):y

Would you like ASAP to generate alerts as EMS events(N):y

What should the ASAP rate be, in minutes(5):3

ASAP maintains a historical database that it can purge every
night or it can rename the files each night thereby preserving
data.

Would you like ASAP to retain the data(N):y

Starting with release 2.05 and later ASAP can automatically
maintain the number of days of history it will retain. You can
determine how much space this will require by using the ASAP
Database Calculator at http://NonstopAsap.com/Contributions.htm.
You can also monitor database usage with the ASAP command
interpreter DB command.
```

If you do not allow ASAP to automatically retain data for a certain number of days, it is up to the user to manage the accumulated data so it doesn't fill up the disk drive.

Do you want ASAP to automatically retain data for a number of days(Y)?y

What is the default number of days you would like to retain(7)?

Would you like to specify the number of days to save per entity(Y)?n

If this instance of ASAP will be used to monitor multiple nodes, then it is important to synchronize time across all nodes in the collection.

Would you like ASAP to synchronize time on the nodes(N):y

ASAP can automatically enter actions against objects to recover from failure conditions.

Would you like ASAP to enter actions to recover failed objects(Y)?y

The ASAP Extension is required for ASAP to monitor applications.

Would you like ASAP to monitor APPLICATIONS(N):y

Would you like ASAP to monitor the CIP subsystem(N):

Would you like ASAP to monitor COMM lines(Y):

Would you like ASAP to monitor DISK devices(Y):

Would you like ASAP to monitor EXPAND lines(Y):

Would you like ASAP to monitor FILES(Y):

Would you like ASAP to monitor EXPAND nodes(N):

Would you like ASAP to monitor PROCESSES(Y):

Would you like ASAP to monitor RDF(N):y

Would you like ASAP to monitor SPOOLER subsystems(Y):

Would you like ASAP to monitor KMSF Swap files(Y):

Would you like ASAP to monitor TAPE devices(Y):

Would you like ASAP to monitor TCP/IP and Telserv(N):

Would you like ASAP to monitor TMF(Y):

Please enter a subvolume for the new ASAPCONF file(\$SYSTEM.SYSTEM):\$asap.testdb2

Creating \$ASAP.TESTDB2.ASAPCONF

Would you like the Wizard to restart ASAP using the new configuration(Y)?n

You may now run ASAP CI and enter MONITOR FILE and MONITOR PROCESS commands to configure the files and processes that ASAP will monitor.

The ASAP Wizard has successfully completed.

2>

ASAP Object Files

The ASAP system consists of these files on \$SYSTEM.SYSTEM:

Object File	Full Name	Description
ASAP	ASAP Conversational Interface	Starts and manages the Monitor and Collector programs, and displays the ASAP availability information.
ASAPCIP	CIP Statistics Gathering process	Gathers availability information from Cluster I/O Modules on NonStop™ servers.
ASAPCMM	COMM Statistics Gathering process	Gathers availability information for every monitored SWAN line on the node.
ASAPCOL	Statistics Collect Server process	Runs continually and manages the ASAP database.
ASAPCONF	ASAP Configuration File	Obey file for setting up the ASAP environment.
ASAPDSK	Disk Statistics Gathering Process (SGP)	Gathers availability information for every monitored disk device on the node.
ASAPEXP	Expand Statistics Gathering Process (SGP)	Gathers availability information for every monitored Expand line on the node.
ASAPFIL	File Statistics Gathering Process (SGP)	Gathers availability information for every user selected file or user-selected subvolume on the node.
ASAPMON	ASAP Monitor process	Gathers availability information for every monitored CPU and busiest process.
ASAPNCP	Node Statistics Gathering Process (SGP)	Gathers path information and traffic statistics between two Expand nodes.
ASAPPRO	Process Statistics Gathering Process (SGP)	Gathers availability information for every user-selected process on the node.

ASAPRDF	RDF Statistics Gathering Process (SGP)	Gathers availability information for either the RDF Extractor or RDF Updater on the node.
ASAPSPL	Spooler Statistics Gathering Process (SGP)	Gathers availability information for every monitored spooler subsystem on the node.
ASAPSWP	Swap Statistics Gathering Process (SGP)	Gathers availability information for every monitored KMSF swapfiles on the node.
ASAPTAP	Tape Statistics Gathering Process (SGP)	Gathers availability information for every monitored tape device on the node.
ASAPTCP	TCP Statistics Gathering Process (SGP)	Gathers availability and statistics information for nine TCP/IP entities and three Telserv entities.
ASAPTMF	TMF Statistics Gathering Process (SGP)	Gathers availability information for the TMF subsystem on the node.
ASAPXSGP	ASAP Statistics Gathering Process (SGP)	Gathers service-level metrics about customer applications. It is an optional product, part of the ASAP Extension product (ASAPX).

These descriptions of the ASAP object files explain which ASAP components are required on various nodes of the network:

ASAP Object File

The object file ASAP is the ASAP Conversational Interface (CI), which starts and manages the Monitor and Collector processes, and displays the collected ASAP availability information. The ASAP CI provides command and control of the ASAP environment and enables interactive reporting of availability information. STATUS, STARTUP, and SHUTDOWN commands control Monitor and Collect processes. Use the SET command to set up options for the Monitor, Collector, and the Statistic Gathering processes. APP, CPU, DISK, FILE, LH, NODE, PROCESS, PROCSSBUSY, RDF, SPOOLER, SYSTEM, TAPE and TMF commands provide ad hoc statistics reports. For the syntax and detailed descriptions of these commands, see [Section 6: ASAP Conversational Interface Commands](#).

ASAPCIP Object File

The ASAP CIP Statistics Gathering (ASAPCIP) Process collects availability information for Cluster I/O Modules on NonStop™ servers, including availability, usage, performance, and connection information. The ASAP Monitor starts the CIP SGP on its node if CIP is enabled. See the [CIP command](#).

ASAPCMM Object File

The Monitor starts the COMM SGP on its node if Comm is enabled. The ASAP Statistics Gathering (ASAPCMM) Process collects availability information for all monitored SWAN communication line types. ASAPCMM uses the Measure programmatic interface to obtain raw communication line counters from the communication processes. For a complete list of COMM availability counters, see the [COMM Command](#).

ASAPCOL Object File

The ASAP database includes current and historical information about system performance and operational status of system objects. It is automatically created when the ASAP Collect process is first started. The Collect process receives statistics and status messages from ASAP monitors throughout the network, parses the messages, and stores the information in the normalized ASAP database.

The Collect process automatically maintains the database. The Collect process includes database maintenance options to specify how much history to retain. You can configure database history to only retain data since the last cleanup time; for example, every night at midnight. Or you can configure it to retain ASAP history for a longer period of time, or indefinitely. For more information about database maintenance options, see the SET RETAIN option under the [SET Command](#).

The ASAP Collect process is continuous. If a CPU fails, the backup Collect process automatically takes over so database access is always available. The ASAP database is a real-time normalized database and can be queried online. The ASAP product subvolume includes Enform queries that allow batch-type queries of network performance and operational status.

The statistics Collect server (ASAPCOL) process runs continually and manages the ASAP database. It:

- Creates and maintains ASAP structured files.
- Maintains statistics sent to the Collect process from various Monitor processes.
- Manages ongoing space management of active database files.

If you manually execute the ASAPCOL object file from a TAQL prompt with a process name of \$APPDB, \$CPUDB, \$DSKDB, \$NETDB, \$SYSDB, \$PCBDB, \$EXPDB, \$FILDB, \$PRODB, \$RDFDB, \$SPLDB, \$TAPDB, or \$TMFDB, it acts as an Enform server rather than a Collect process. For information on how to use ASAPCOL as an Enform server, see the ENFALL file in the ASAP distribution subvolume.

ASAPCONF File

The ASAPCONF file contains configuration information for the ASAP and ASAP Extension subsystems. Before trying to start the ASAP and ASAPX subsystems, make sure that they are properly installed.

ASAPDSK Object File

The ASAP Monitor starts the Disk SGP on its node if Disk is enabled. The ASAP Statistics Gathering (ASAPDSK) Process collects availability information for the ASAP for all nonphantom disk devices. ASAPDSK uses the Measure Programmatic Interface to obtain raw disk counters from the disk process. For a complete list of Disk availability counters, see the [DISK Command](#).

ASAPEXP Object File

The ASAP Monitor starts the Expand SGP on its node if Expand is enabled. The ASAP Statistics Gathering (ASAPEXP) Process collects availability information for the Expand line handlers. The utilization counters gathered include page faults, BCC, Naks sent and received, and packets sent and received. For a complete list of the Expand availability counters, see the [LH Command](#). The Expand SGP communicates directly with the Expand Manager process to get Expand availability information.

ASAPFIL Object File

The ASAP Monitor starts the File SGP on its node if File is enabled. The ASAP Statistics Gathering (ASAPFIL) Process collects availability information for user-selected files or subvolumes. The utilization counters gathered include percent file full, number of files in the subvolume, and number of partitions. For a complete list of file availability counters, see the [FILE Command](#).

ASAPMON Object File

The ASAP node Monitor (ASAPMON) process manages all node-level processing for the network node on which it is running, and creates the Statistics Gathering processes (SGPs) in each CPU of the node where it is located. These SGPs collect CPU availability information and PROCESSBUSY availability information for the CPU in which it resides. System availability is the average availability for all the monitored CPUs on the node. The ASAP Monitor monitors CPU up and down messages, stop or abend messages, and creates the SGPs according to the changing status of the CPUs. The ASAP Monitor also starts the node entity SGPs, APP, Disk, File, Node, Process, RDF, Spooler, Tape and TMF.

The ASAPMON processes use special NonStop™ OS procedures to collect CPU, Processbusy, and System Availability information.

This object file is required on all monitored nodes.

When the ASAP Conversational Interface (CI) starts a remote ASAP Monitor process, the Monitor starts various monitoring functions specified in the ASAPCONF file or from interactive user commands. The ASAP Monitor is responsible for starting and maintaining the proper monitoring software in its node. If any Monitor or Statistics Gathering process (SGP) goes down for any reason, the ASAP Monitor in that node coordinates an automatic restart of processes as needed. For example, if a CPU fails and comes back online, the CPU SGP for that CPU is restarted. Or, if a remote ASAP Monitor process fails, its backup process takes over since the ASAP Monitor process is continuous.

The remote ASAP Monitor starts the CPU SGP in each CPU of the monitored node. CPU SGPs are responsible for gathering CPU-level statistics, including CPU performance metrics, Operational state, and information about the busiest processes in their CPU. The CPU SGPs report CPU-level information to the ASAP Monitor process. The Monitor bundles this information into a single message and forwards it to the ASAP Collect process on the main Collection node. The Collect process parses this information and stores it in the normalized ASAP database. The data can be displayed with the ASAP CPU and PROCESSBUSY commands, or with various block-mode or workstation programs.

ASAPNCP Object File

The ASAP Monitor starts the Node SGP on its node if the Node SGP is enabled. The ASAP Statistics Gathering (ASAPNCP) Process collects statistics for the Expand end-to-end node statistics. The statistics gathered are path information statistics and traffic statistics between two Expand nodes. ASAPNCP communicates directly with the Expand Manager process to obtain raw counters. For a complete list of node availability counters, see the [NODE Command](#).

The Node SGP does not support monitoring or Discrete Objectives Thresholds (DOTs).

ASAPPRO Object File

The ASAP Monitor starts the Process SGP on its node if Process is enabled. The ASAP Statistics Gathering (ASAPPRO) Process collects availability information for the user-selected processes. The utilization counters gathered include process busy time, process queue length, and process wait time. The Process SGP gets availability information through the Guardian API. For a complete list of Process availability counters, see the [PROCESS Command](#).

ASAPRDF Object File

The ASAP Monitor starts the RDF SGP on its node if RDF is enabled. The ASAP Statistics Gathering (ASAPRDF) process collects availability information for the RDF updater or RDF extractor. The utilization counters gathered include relative time delay,

Audit trail index, and Audit volume. For a complete list of RDF availability counters, see the [RDF Command](#).

ASAPSPL Object File

The ASAP Monitor starts the Spooler SGP on its node if Spooler is enabled. The ASAP Statistics Gathering (ASAPSPL) Process collects availability information about spooler subsystems on the node. ASAPSPL uses the spooler procedure call interface to communicate with the spooler supervisors to get information about the spooler, collectors, devices, and print processes. The utilization counters gathered include Spooler percent full, number of device locations, and number of jobs currently printing. For a complete list of Spooler availability counters, see the [SPOOLER Command](#)

ASAPSWP Object File

The ASAP Monitor starts the Swap SGP on its node if the Swap SGP is enabled. The ASAP Statistics Gathering (ASAPSWP) Process collects availability information about KMSF swap files. ASAPSWP uses the Seeview Server Gateway for obtaining KMSF status and SWAP usage counters. Usage counters collected for each CPU on the node, including good/failed allocations, good/failed reservations for process creations, available swap space which is the total of all swapfiles for the CPU and more. For a complete list of Swap availability counters, see the [SWAP Command](#).

ASAPTAP Object File

The ASAP Monitor starts the Tape SGP on its node if the Tape SGP is enabled. The ASAP Statistics Gathering (ASAPTAP) Process collects availability information about tape devices for the ASAP Tape entity. ASAPTAP communicates with the DSM/Tape Catalog MediaSRV server using the SPI interface to obtain the status and information about the tape devices and tape mount requests on the node. The utilization counters gathered include number of outstanding tape mounts and Label mode checks. For a complete list of Tape availability counters, see the [TAPE Command](#).

ASAPTCP Object File

The ASAP TCP/IP Statistics Gathering (ASAPTCP) process collects availability information for sixteen separate TCP/IP entities and three Telserv entities, including availability, usage, performance, and connection information. The ASAP Monitor starts the TCP SGP on its node if TCP is enabled. See the [TCP Command](#)

ASAPTMF Object File

The ASAP Monitor starts the TMF SGP on its node if the TMF SGP is enabled. The ASAP Statistics Gathering (ASAPTMF) Process collects availability information about the TMF subsystem. Valid TMF domains are: Transactions, AuditTrail, AuditDump, and TMF for the TMF subsystem itself. The utilization counters gathered include

percentage of audit trail used, transaction rate of the TMF subsystem, and number of pending transactions. For a complete list of TMF availability counters, see the [TMF Command](#). ASAPTMF communicates directly with the TMFServe server process for obtaining TMF availability information.

ASAPXMON Object File

The optional ASAPX Monitor process (ASAPXMON) allocates and manages shared memory in each processor and performs sampling and metric computation.

Applications use the ASAP_REGISTER_ procedure to communicate with ASAPXMON. ASAPXMON allocates shared memory for the application and returns the memory address used in subsequent calls to update procedures.

At each interval, ASAPXMON processes sample shared memory and computes statistics for each application domain.

ASAPXMON processes are started and managed by ASAPXSGP.

ASAPXSGP Object File

The optional ASAPX Statistics Gathering Process (ASAPXSGP) initiates statistics gathering at each interval and starts and manages ASAPXMON processes.

In normal operating mode, the SGP gathers statistics from all ASAPXMON processes and forwards them to the ASAP Collector process. The SGP also performs aggregate calculations across all ASAPXMON processes and creates any necessary aggregate domain records.

In Write-to-Collector mode, each ASAPXMON writes directly to an ASAP Collector process and only sends aggregate total records to the SGP.

ASAPXSGP is a standard ASAP SGP and is started and managed by ASAPMON.

The ASAPX Library

The optional ASAPX Library is made of four object files:

Object File	Description	Installation
ASAPXDLL	A dynamic link library (DLL) containing the ASAPX API procedures	Installed by the ASAPX installation process.
ASAPXLIB	A TNS user-library containing the ASAPX API procedures	Installed by the ASAPX installation process
ASAPXSRL	A private Shared Runtime Library containing the ASAPX API procedures	Installed by the ASAPX installation process
ASAPXSRO	A native linkable object for binding the native API into an existing private Shared Runtime Library	Can be found in the ASAPX ISV subvolume

For more information about the ASAPXSGP, ASAPXMON, and ASAPX Library, see the ASAP Extension Manual.

ASAPGATE

The optional ASAP Hybrid for NonStop™ Server Gateway (ASAPGATE) process provides gateway functions between Linux systems running ASAP Hybrid for Linux and the NonStop™ server.

ASAPPXY

The optional ASAP Hybrid for NonStop™ Server Proxy (ASAPPXY) process acts as a proxy Statistics Gathering Process for ASAP Hybrid. The ASAP Monitor starts the optional Proxy SGP on its node if Proxy is enabled.

For more information about the ASAPPXY and ASAPGATE object files, see the *ASAP Hybrid Manual*.

Size the ASAP Database Subvolume

With each new release of ASAP, you must update the database subvolume with the latest version of the installation subvolume files, along with the DDL data dictionary in that subvolume:

- If you ran the **ASPSETUP Full Install on this Node** option, the database was updated automatically.
- If you did not run the **ASPSETUP Full Install on this Node** option, use the **ASPSETUP Database subvol Install only** option.

The **Database subvol Install only** option looks identical to the **Full Install on this Node** option, but it only creates the database subvolume location and compiles the database DDL.

To determine the approximate daily size of the database at a central location, use this formula:

Bytes =	Cpus	* 132	* (1440/rate)
	+ Nodes	* 132	* (1440/rate)
	+ Apps	* 556	* (1440/rate)
	+ CIP	* 540	* (1440/rate)
	+ Comm	* 238	* (1440/rate)
	+ Disks	* 224	* (1440/rate)
	+ Expand	* 404	* (1440/rate)
	+ Files	* 220	* (1440/rate)
	+ Ncpnodes	* 3630	* (1440/rate)
	+ Processes	* 312	* (1440/Rate)
	+ ProcessBusyEntries * Cpus	* 174	* (1440/rate)
	+ RDFs	* 230	* (1440/Rate)
	+ Spoolers * 4	* 212	* (1440/Rate)

+ Swapfiles * Cpus	* 1548	* (1440/Rate)
+ Tapes	* 162	* (1440/Rate)
+ TCP	* 1142	* (1440/Rate)
+ TCPPort	* 246	* (1440/Rate)
+ TCPRoute	* 140	* (1440/Rate)
+ TCPSubnet	* 176	* (1440/Rate)
+ Telserv	* 182	* (1440/Rate)
+ TelservService	* 158	* (1440/Rate)
+ TelservWindow	* 370	* (1440/Rate)
+ TMFs * 4	* 202	* (1440/Rate)

Where:

Cpus	= Total CPUs monitored on all nodes.
Nodes	= Total number of nodes monitored.
Apps	= Total application domain names monitored on all nodes.
CIP	= Total number of CIP CLIMs, Monitors and Providers.
Comm	= Total SWAN comm lines monitored on all nodes.
Disks	= Total disk volumes monitored on all nodes.
Expand	= Total Expand lines monitored on all nodes.
Files	= Total file domains monitored on all nodes.
Ncpnodes	= Total NCP nodes monitored using the SET NODE ON option.
Processes	= Total processes monitored on all nodes.
ProcessBusyEntries	= Number of busiest processes collected per CPU.
RDFs	= Total RDF systems monitored on all nodes.
Spoolers	= Total spoolers monitored on all nodes.
Swapfiles	= Total swapfiles monitored on all nodes.
Tapes	= Total tapes monitored on all nodes.
TCP	= Total TCPICMP, TCPIP, TCPProcess, TCPQIO, TCP RTE data (combined) on all nodes.
TCPPort	= Total number of TCP/IP monitored and active ports on all nodes.
TCPRoute	= Total number of TCP/IP Routes monitored on all nodes.
TCPSubnet	= Total number of TCP/IP Subnets monitored on all nodes.
Telserv	= Total number of Telserv processes monitored on all nodes.
TelservService	= Total number of Telserv services monitored on all nodes.
TelservWindow *	= Total number of monitored and active Telserv services on all nodes.
TMFs	= Total TMF systems monitored on all nodes.
Rate	= Sample interval (in minutes).

Example

For an ASAP configuration that monitors:

- 80 CPUs
- 10 nodes
- 100 disk volumes

- 30 Expand line-handlers
- 50 files
- 50 processes
- 3 ProcessBusyEntries per CPU

The daily storage requirement for a five-minute sample interval requires approximately 32 megabytes of disk space:

Bytes	=	80	*	128	*	(1440/5)	-- CPU	2,949,120	
		10	*	128	*	(1440/5)	-- Node	+ 368,640	
		100	*	222	*	(1440/5)	-- Disk	+ 6,393,600	
		30	*	400	*	(1440/5)	-- Expand	+ 3,456,000	
		50	*	212	*	(1440/5)	-- File	+ 3,052,800	
		50	*	312	*	(1440/5)	-- Process	+ 4,492,800	
		3	*	80	*	(1440/5)	-- ProcessBusy	+ 12,026,880	
	=	31,587,840							

For a sample interval of ten minutes, the daily storage requirement is reduced to approximately 16 megabytes.

Recommended Sample Rates

The minimum recommended sample rates for a single ASAP Collect process running on a K20000 server in a large network with 4 Expand line-handlers per node and 10 Disk volumes per node both entities enabled are:

- 1 minute for a 25 node network with 2 CPUs per node
- 2 minutes for a 25 node network with 4 CPUs per node
- 3 minutes for a 50 node network with 2 CPUs per node
- 5 minutes for a 100 node network with 4 CPUs per node
- 10 minutes for a 100 node network with 8 CPUs per node

Starting and Stopping ASAP

This subsection explains how to stop and start the ASAP Environment. For frequent use, you can put these commands in OBEY files.

Starting ASAP

- 1 Start ASAP by entering ASAP at a TACL prompt:

```
TACL 1 > ASAP
```

```
ASAP - Availability Stats and Performance - T0402V02.5 (15OCT2004)
```

```
Copyright (C) 1999, 2001-2004 Hewlett-Packard Company
```

```
Obey $SYSTEM.SYSTEM.ASAPCONF
```

```
+
```

- 2 Verify the ASAP configuration by entering the SET command to display the current configuration values. For example:

```
+SET
```

```
ASAP Configuration Control settings:
```

```
Version..... T0402V02.05 (15SEP2004)
Action..... ON
ActObject..... $SYSTEM.SYSTEM.SEEVIEW
ActShell..... $SYSTEM.SYSTEM.ASAPACT
Asaplog..... $SYSTEM.SYSTEM.ASAPLOG
Backupcpu..... *
Buffered..... OFF
Cleantime.....0000
Collect..... \CENTDIV
DB..... $DATA.ASAPDB.DB
DBMaxDays.....0
DBMaxDaysToPurge.....2
DBRolloverSubvolPrefix.. ASAP
DBRolloverFilePrefix.... Z
EDL..... $SYSTEM.SYSTEM.ASAP2EDL
Helpfile..... $SYSTEM.SYSTEM.ASAPHELP
ID..... $V03(Collect $V03S, Monitor $V03M Scp $V03*)
IoControl.....64
Logging..... OFF
Logfile..... $SYSTEM.SYSTEM.LOGFILE
LogInterval.....60
LogSuppress..... OFF
MonitorAdd..... ON
NodeDownAlert..... ON
Objcollect..... $SYSTEM.SYSTEM.ASAPCOL
Objmonitor..... $SYSTEM.SYSTEM.ASAPMON
ObjectivesAudit..... OFF
ObjectivesDB..... $DATA5.DOTS.OBJDB
ObjectivesEvent..... ON
ObjectivesEventConsole. ON
ObjectivesEventConsoleUp ON
ObjectivesEventMax. 20
ObjectivesEventSubject. OFF
ObjectivesEventState.... WARNING
ObjectivesEventUp..... ON
ObjectivesRank..... STANDARD
Primary..... *
Priority.....160
```

```

ProcessBusy..... ENTRIES 3, SUBSAMPLES 3
Rate.....1
RemoteSecurity..... ON
Retain..... NONE
Sleeptime..... 0000,0000
Sync..... 0,0
SyncSlow..... OFF
Extended Entity Information:
APP.....ON
  Object..... $SYSTEM.SYSTEM.ASAPXSGP
  Parameters..... "CONFIG \CENTDIV.$system.system.ASAPXCNF"
  CPU..... *
CIP.....ON
  Object..... $SYSTEM.SYSTEM.ASAPCIP
  Parameters..... "SCPNAME $ZNES "
  CPU..... *
COMM.....ON
  Object..... $SYSTEM.SYSTEM.ASAPCMM
  Parameters.....
  CPU..... *
DISK.....ON
  Object..... $SYSTEM.SYSTEM.ASAPDSK
  Parameters.....
  CPU.....15
EXPAND.....ON
  Object..... $SYSTEM.SYSTEM.ASAPEXP
  Parameters.....
  CPU..... *
FILE.....ON
  Object..... $SYSTEM.SYSTEM.ASAFIL
  Parameters.....
  CPU.....3
NODE.....ON
  Object..... $SYSTEM.SYSTEM.ASAPNCP
  Parameters.....
  CPU..... *
PROCESS.....ON
  Object..... $SYSTEM.SYSTEM.ASAPPRO
  Parameters.....
  CPU.....9
RDF.....ON
  Object..... $SYSTEM.SYSTEM.ASAPRDF
  Parameters.....
  CPU.....0
SPOOLER.....ON
  Object..... $SYSTEM.SYSTEM.ASAPSPL
  Parameters.....
  CPU.....0
SWAP.....ON
  Object..... $SYSTEM.SYSTEM.ASAPSWP
  Parameters.....
  CPU.....3
TAPE.....ON
  Object..... $SYSTEM.SYSTEM.ASAPTAP
  Parameters.....
  CPU.....1
TCP.....ON

```

```

Object..... $SYSTEM.SYSTEM.ASAPTCP
Parameters..... "RATE 5,SCPNAME $ZNES,PORT,WINDOW,SERVICE"
CPU.....2
TMF.....ON
Object..... $SYSTEM.SYSTEM.ASAPTMF
Parameters.....
CPU.....2

```

3 If the settings are not correct, do one of:

- Exit the ASAP Command Interpreter and change the ASAPCONF file with the correct settings; then restart ASAP as shown in Step .
- Override the defaults by entering the appropriate SET commands.
- Create a custom OBEY file that includes the SET commands you want.

4 Once the SET parameters are correct, start the Collector database server:

```
+STARTUP COLLECT      !Starts the DB Collector process on the local node
```

5 After successfully starting the ASAP Collector, start the Monitor processes on the nodes that are to report to the Collector that was just started. For example:

```
+STARTUP              !Starts Monitor on the local node
```

```
+STARTUP \NY         !Starts Monitor in New York
```

```
+STARTUP \CHICAGO    !Starts Monitor in Chicago
```

6 If you are using the ASAP Client, you can start it at this time. For more information, see the ASAP Client Manual.

ASAP Status

The ASAP STATUS command obtains the status of the Collector or Monitor processes. To obtain the overall status of the Collect and Monitor processes, enter:

```
+STATUS COLLECT      !Status of the Collector on the local node
```

```
+STATUS \NY         !Status of the Monitor in New York
```

```
+STATUS              !Status of the local node
```

Stopping ASAP

To stop the Collector or Monitor processes:

1. Use the ASAP SHUTDOWN command.
2. After using the SET command on nodes currently executing the Collector or Monitor processes, stop and restart the processes so that the configuration changes take effect.

For example, if a node is currently collecting Expand statistics and you no longer require this type of statistic, but you still require CPU and ProcessBusy statistics, enter SET EXPAND OFF.

3. Shut down the Monitor process on the node.
4. Once successfully stopped, use the ASAP STARTUP command to restart the node.

The CPU and ProcessBusy statistics are collected, but the Expand Line-handler statistics are no longer reported to the Collection node.

Examples of stopping Monitors or Collectors are:

```
+SHUTDOWN \CHICAGO      !Shuts down the Monitor in Chicago
+SHUTDOWN \NY           !Shuts down the Monitor in New York
+SHUTDOWN               !Shuts down the local node
+SHUTDOWN \*            !Shuts down all the Monitors in the network
+SHUTDOWN COLLECT       !Shuts down the Collector process on the local node
+SHUTDOWN \CHICAGO COLLECT !Shuts down the Collector process on the Chicago
                        node
```

To stop the ASAP session and return to the TACL prompt, type EXIT and press Return.

Using OBEY Files

You can set up OBEY files using the editor of your choice. To execute the OBEY file using the ASAP CI, enter OBEY *\$vol.subvol.filename*. The SHUTDOWN commands can be placed in Edit type file and then obeyed from within the ASAP CI.

To create OBEY files while in the ASAP CI, use the SET LOGGING ON command and the SET LOGFILE *\$vol.subvol.filename*. When you use these two commands, all commands that you have entered are written to the file specified with the LOGFILE command. You can then enter the commands for later use. When you use an OBEY file as a run parameter, the OBEY file is executed, but you do not get the ASAP prompt. After executing the commands, control returns to the TACL prompt.

Section 3: Running ASAP

ASAP is started from a single START command, and is self-configuring in terms of what objects it automatically monitors when the configuration file is set up correctly and executed. It also includes options to modify their default configurations. For more information, see [Section 6: ASAP Conversational Interface Commands](#).

To start the ASAP Conversational Interface (CI):

- 1 Type `ASAP` at a TACL prompt. Whenever the ASAP CI is started, it automatically obeys the ASAPCONF configuration file. If SET APP ON is contained in the ASAPXCNF file, ASAP/ASAPX interoperation is assured.
- 2 Type `start collect` to start the ASAP database Collect process.
- 3 Type `start \node` to start the remote ASAP and ASAPX Monitor processes for each node you want to monitor.

Topic	Page
Starting the ASAP Collector	76
Starting the ASAP Monitors and SGPs	78
Getting Collector and Monitor Status Information	80
Starting ASAP With the Default Configuration	82
Changing the ASAP Configuration Dynamically	82
Viewing Statistics	82
Shutting Down the ASAP Collector and ASAP Monitors	83
Synchronizing the Time of Day on Selected Nodes	83
Monitoring Different Sets of Network Nodes	84
Viewing the ASAP Event Log	85
ASAP Help	86

Starting the ASAP Collector

Before you start the ASAP Collector, you must use the ASAP SET command to define the ASAP Collector options. You can set options through OBEY files, TACL macros, or interactively from the ASAP CI. For a complete list of SET options for the collector, see the [SET Command](#).

When you set up the ASAP Collector, only these SET command options affect its configuration:

- SET BACKUPCPU

- SET BUFFERED
- SET CLEANTIME
- SET DB
 - SET DBMAXDAYS
 - SET DBMAXDAYSTOPURGE
 - SET DBROLLOVERSUBVOLPREFIX
 - SET DBROLLOVERFILEPREFIX
- SET ID
- SET OBJCOLLECT
- SET PARTITION
- SET PRIMARY
- SET PRIORITY
- SET RATE
- SET RETAIN
- SET SLEEPTIME

After you define the ASAP Collector options, you can start the Collector process. This example sets up the collector on \$Data.asapdb, purging all records daily at 12:15 a.m. and not collecting availability records between 11:00 p.m. and 12:30 a.m.:

```
+set db $data.asapdb.db
+set retain none
+set sleeptime 2300, 0030
+set clean 0015
+STARTUP COLLECT
```

From the same ASAP CI, another collector on \NY can be set up to collect on \$Data2.asapdb, purge all records daily at 2:15 a.m., and not collect availability records between 10:00 p.m. and 2:30 a.m.

```
+set db $data2.asapdb.db
+set retain none
```

```
+set sleeptime 2200, 0230
+set clean 0215
+STARTUP \NY COLLECT
```

Starting the ASAP Monitors and SGPs

Before you start the ASAP Monitors and SGPs, define the ASAP Monitor and SGP options using the ASAP SET command. Options are set through OBEY files, TACL macros, or interactively from the ASAP CI. When you set up the SGP or the Monitor, only these SET command options affect their configurations:

- SET APP
- SET BACKUPCPU
- SET CIP
- SET COLLECT
- SET COMM
- SET DISK
- SET EXPAND
- SET FILE
- SET ID
- SET NODE
- SET OBJECTIVESAUDIT
- SET OBJECTIVESDB
- SET OBJECTIVESEVENT
- SET OBJECTIVESEVENTUP
- SET OBJECTIVESEVENTCONSOLE
- SET OBJECTIVESEVENTCONSOLEUP

- SET OBJECTIVESEVENTSTATE
- SET OBJECTIVESEVENTSUBJECT
- SET OBJECTIVESEVENTMAX
- SET OBJECTIVESRANK
- SET OBJMONITOR
- SET PRIMARY
- SET PRIORITY
- SET PROCESS
- SET PROXY
- SET PROXYOBJECT
- SET PROXYPARAM
- SET RATE
- SET RDF
- SET SPOOLER
- SET SWAP
- SET SYNC
- SET SYNCSTLOW
- SET TAPE
- SET TCP
- SET TMF

For more information about these commands, see the [SET Command](#).

Starting the Monitor Processes

After you define the ASAP Monitor and SGP options, you can start the Monitor processes. This example sets up the ASAP subsystem to monitor two nodes. Disk and Expand objects are being monitored on both nodes, and application (APP) objects are being monitored on one of the nodes, with an availability rate of five minutes:

```
+SET OBJECTIVESEVENTS ON
+SET DISK ON, PARAMETERS "CPU 2 VOLUME $DATA"
+SET APP ON, PARAMETERS "CONFIG $DATA2.APP.ASAPXCNF"
+SET EXPAND ON
+SET RATE 5
+STARTUP \NY
+SET APP OFF
+STARTUP \LA
```

Getting Collector and Monitor Status Information

The ASAP STATUS command obtains the status of the Collector or Monitor processes. The ASAP STATUS command has no options. To obtain the overall status of the Collect process, enter:

```
+STATUS COLLECT          !Status of the Collector on the local node

\CENTDIV  Primry Backup Pri Access  Reqs  Lmsgs Created  ReqTime  SrvTime  Et
-----
$ZOOS V03  1,174  0,080 160 255,255 11519      8  6/20 16 1216:23 1216:23  0
$ASAP.ASAPDB.DBAPP Open(0) Days NONE
$ASAP.ASAPDB.DBCOM Open(0) Days 30
$ASAP.ASAPDB.DBCPU Open(0) Days 365
$ASAP.ASAPDB.DBDSK Open(0) Days 7
$ASAP.ASAPDB.DBEXP Open(0) Days 7
$ASAP.ASAPDB.DBFIL Open(0) Days 7
$ASAP.ASAPDB.DBNCP Open(1) Days 7
$ASAP.ASAPDB.DBPRO Open(0) Days 7
$ASAP.ASAPDB.DBPCB Open(0) Days 30
$ASAP.ASAPDB.DBRDF Open(0) Days NONE
$ASAP.ASAPDB.DBSPL Open(0) Days 30
$ASAP.ASAPDB.DBSWP Open(0) Days 10
$ASAP.ASAPDB.DBSYS Open(0) Days 365
$ASAP.ASAPDB.DBTAP Open(0) Days 7
$ASAP.ASAPDB.DBTMF Open(0) Days 7
(rate 1, sleep 0000,0000, clean 0000, retain ROLLOVER, DBMaxDays 1)
```

The STATUS command returns the Collector's node name, the process ID and version, the database files, and the status of the DB files. The value in parentheses indicates the last Guardian file error against the file. The options for the Collector process are also displayed.

The display returned from the STATUS command is defined in the following table.

Table 3-1. STATUS COLLECTOR Output

Column Heading	Definition
\$ZOOS V02	Identifies the Collector process and version
Primry	Identifies the process primary CPU,PIN
Backup	Identifies the process backup CPU,PIN
Pri	Indicates the process execution priority
Access	Indicates the process security ID
Reqs	Indicates the number of requests this process has received since it was started
Lmsgs	Indicates the number of log messages this process has logged
Created	Indicates the month/day/hour that this process was started
ReqTime	Indicates the time in hours/minutes/seconds that this status request was received by each process
SrvTime	Indicates the time in hours/minutes/seconds that this status request was replied to by each process
ET	Indicates the elapsed time for this process to reply to the status request

To get the overall status of the Monitor process on a specific node, enter:

```
+STATUS
\NY
\NY          !Status of the Monitor in New York
Primry Backup Pri Access Reqs Lmsgs Created ReqTime SrvTime Et
-----
$ZOOM V05 1,215 0,153 160 255,255 118 5 3/17 19 2135:17 2135:17 0
$ZOOM0 V05 0,233 160 255,255 322 1 3/17 19 2135:17 2135:17 0
$ZOOM1 V05 1,115 160 255,255 322 1 3/17 19 2135:17 2135:17 0
$ZOOM2 V05 2,084 160 255,255 322 1 3/17 19 2135:17 2135:17 0
$ZOOM3 V05 3,202 160 255,255 322 1 3/17 19 2135:17 2135:17 0
$ZOON V04 0,230 RDF 160 255,255 117 1 3/17 19 2135:17 2135:17 0
$ZOOG V05 1,039 NCP 160 255,255 117 1 3/17 19 2135:17 2135:17 0
$ZOOH V04 1,151 APP 160 255,255 117 2 3/17 19 2135:17 2135:17 0
$ZOOJ V05 1,178 DSK 160 255,255 116 2 3/17 19 2135:17 2135:17 0
$ZOOK V05 1,092 EXP 160 255,255 117 1 3/17 19 2135:17 2135:17 0
$ZOOL V05 1,095 FIL 160 255,255 117 1 3/17 19 2135:17 2135:17 0
$ZOOO V05 1,191 SPL 160 255,255 117 1 3/17 19 2135:17 2135:17 0
$ZOOQ V05 1,181 TAP 160 255,255 117 1 3/17 19 2135:17 2135:17 0
$Zoor V05 1,121 TMF 160 255,255 117 1 3/17 19 2135:17 2135:17 0
$ZOOT V05 1,132 PRO 160 255,255 117 1 3/17 19 2135:17 2135:17 0
$ZOOU V05 1,194 COM 160 255,255 116 2 3/17 19 2135:17 2135:17 0
$ZOOV V05 2,114 TCP 160 255,255 32 1 3/17 19 2135:17 2135:17 0
$ZOOA V05 1,091 SWP 160 255,255 117 1 3/17 19 2135:17 2135:17 0
$ZOOC V04 1,091 CIP 160 255,255 117 1 3/17 19 2135:17 2135:17 0
(collect \NY.$ZOOS, rate 1)+
```

The STATUS command returns the node name, the process ID and version, and all of the SGPs that are running on the node. In this example, the SGPs are \$ZOO1 and \$ZOOJ. The options in parentheses for the Monitor process are also displayed.

You can also use the ASAP LOG command to get more node status information. For example:

```
LOG \CHICAGO
```

Starting ASAP With the Default Configuration

When the Monitor starts, it starts the SGPs that were set to ON. The SGPs get the list of domains they should monitor from the Objectives Database. Each SGP configures some domains by default when no entity records are found in the Objectives Database. For more information, see [MONITOR Usage Considerations](#).

Changing the ASAP Configuration Dynamically

You can dynamically modify the ASAP configuration once the SGPs start.

For example, to add a new File domain, set one global File objective, one specific File objective, specify EMS event generation for both objectives, add a new Process domain, configure a specific objective with EMS event generation specification, then commit the changes:

```
+MONITOR FILE $SYSTEM.SYSTEM.USERID, ADD
+RANK FILE, FULL < 90 CRITICAL
+RANK FILE $SYSTEM.SYSTEM.USERID, owner="255,255" CRITICAL REPEAT
+MONITOR PROCESS $ZOOS, ADD
+RANK PROCESS $ZOOS, PRI >= 160 CRITICAL REPEAT
+COMMIT
```

After the COMMIT command executes, the changes appear at the next sampling interval for each entity. You can also dynamically DELETE domains or objectives, then use the COMMIT command to activate those changes. For command descriptions, see [Section 6: ASAP Conversational Interface Commands](#).

Viewing Statistics

Although the regular user interface to view ASAP statistics is the ASAP Client, you can use the ASAP CI can display those statistics by using the entity name as the command name. For example, to view statistics for the DISK, PROCESSBUSY and SPOOLER entities:

```

+DISK
<statistics output for disk on local system>

+DISK \REMOTE
<statistics output for disk on system \REMOTE>

+PB
<statistics for processbusy on local system>

+SPOOLER
<statistics output for spooler on local system>

```

There are many command options for each entity command. For more information, see [Section 6: ASAP Conversational Interface Commands](#).

Shutting Down the ASAP Collector and ASAP Monitors

The SHUTDOWN command stops the Collector or Monitor processes. The Monitor also shuts down all executing SGPs in each CPU. Stop the remote monitors first, then the collection node's Monitor, and finally the Collector process. This example shows the ASAP SHUTDOWN command:

```

+SHUTDOWN \CHICAGO      !Shuts down the Monitor in Chicago
+SHUTDOWN \NY           !Shuts down the Monitor in New York
+SHUTDOWN               !Shuts down the local node
+SHUTDOWN \*           !Shuts down all the Monitors in the network
+SHUTDOWN COLLECT      !Shuts down the Collector process on the local node
+SHUTDOWN \CHICAGO COLLECT !Shuts down the Collector process on the Chicago
                        node

```

Synchronizing the Time of Day on Selected Nodes

ASAP can synchronize the time of day between selected nodes. This is useful when you want to synchronize a monitored node's internal Greenwich Mean Time (GMT) to the GMT of the Collection node. For example:

```

+SYNC \CHICAGO ! sync \CHICAGO GMT time to the Collection nodes GMT
+SYNC \*       ! sync GMT time for all nodes to the Collection nodes GMT

```

The ASAP Collector process should reside on the node where the system management functions are performed. This node has the correct internal GMT time of day set. Synchronizing to the collection node typically ensures that the nodes throughout the network has the correct time of day. The monitor on the specified node synchronizes that node's time of day with the node where its Collect process resides.

Nodes should have the same GMT so that availability data can be correlated. ASAP maintains all times internally in GMT and then converts them to the local civil time of the Collect server. This lets you view system performance in the system's own local time and ignore time zone differences when correlating statistics.

ASAP's clock synchronization algorithm makes instantaneous time changes if the time difference between the central Collect node and the Remote node is greater than two minutes, and uses slow synchronization only if the time difference between the Central node and the Remote node is less than two minutes. To avoid sudden time changes on the Remote nodes, set the ASAP [SET Command](#) option SET SYNC SLOW ON | OFF to ON to slowly synchronize the time of day between selected nodes.

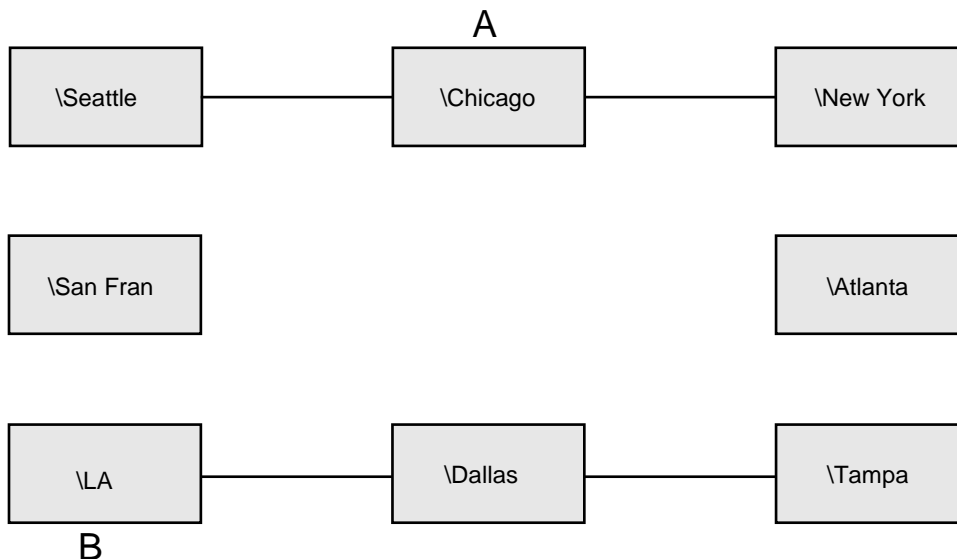
You can set synchronization between nodes to occur automatically with the SET SYNC min, freq option. This option periodically resynchronizes its node's GMT to the Collection node's GMT.

Monitoring Different Sets of Network Nodes

You can configure multiple sets of ASAP Monitor and ASAP Collector processes to operate autonomously with different designated collection locations and options. This lets geographically separate operation groups monitor selectively different sets of network nodes.

The following figure shows an example of such a configuration. Examples A and B illustrate two possible ASAP process configurations.

Figure 3-1. Monitoring Different Sets of Network Nodes



VST001.vsd

Example A

Example A defines a set of ASAP processes that report statistics for all nodes in the network to the Chicago node:

```
SET COLLECT \CHICAGO      !define collection node
SET DB $DATA.ASAPDB.DB   !define database location
SET RATE 15              !define sample interval
SET DISK ON              !Enable disk reporting
SET ID $CHI              !ID for first process set

START \CHICAGO COLLECT   !Start collector on \Chicago
START \*                 !Start monitors & SGPs on all nodes in the network
```

Example B

Example B defines a second set of processes that report statistics for a selected regional group of nodes to the LA node:

```
SET COLLECT \LA          !define collection node
SET DB $DATA2.ASAPDB.DB !define database location on \LA
SET RATE 5               !define sample interval
SET DISK ON              !Enable disk reporting
SET ID $LA              !ID for second process set

START \LA COLLECT       !Start collector on \LA
START \LA               !Start \LA monitor and SGPs
START \SANFRAN          !Start \SanFran monitor and SGPs
START \SEATTLE          !Start \Seattle Monitor and SGPs
START \DALLAS           !Start \Dallas monitor and SGPs
```

Viewing the ASAP Event Log

By default, the Monitor and Collect or processes on each node automatically create and maintain a small log file named \$SYSTEM.SYSTEM.ASAPLOG. In addition, log messages are written to the operator or Viewpoint console, \$0. You can enter the ASAP LOG command to display recent ASAPLOG messages.

ASAP HELP ERROR displays a description of all log error and information message formats. In general, log messages are informational. ASAP should recover from all situations except a total node failure.

This is the output from the ASAP event log file:

Date	Time	Msg#	PID	Message text	Parms
08/02/28	0613:19	1881	ZOOH	INFO ASAPXSGP V03 (15APR2008) started	1
08/02/28	0613:20	1023	ZOOJ	INFO SGP DSK V03 (15APR2008) started	1
08/02/28	0613:22	1881	ZOOH1	INFO ASAPXMON V03 (15APR2008) started	1
08/02/28	0613:22	1881	ZOOH0	INFO ASAPXMON V03 (15APR2008) started	0

```

08/02/28 0613:27 1720 ZOOJ INFO ASAPDSK ZASPZOO started.
08/02/28 0100:29 1008 ZOOM INFO Sync, not required: 0
08/02/28 0100:29 1008 ZOOM INFO Sync, not required: 0

```

The PID column in the INFO display shows the name of the ASAP process. When an ASAP process name ends with an M, it indicates the ASAP Monitor. When it ends with an S, it indicates the ASAP Collector. When it ends with a number or letters A through F, it is a standard SGP. In the previous example, H represents the optional APP SGP, H1 represents the ASAPX Monitor, J represents the Disk SGP, and Params can be Guardian errors or other ASAP error numbers. Report these messages when submitting problem reports against ASAP.

Some common log messages on Remote nodes when the Monitor processes indicate that the Collect process is not available are:

```

08/02/28 1121:01 1001 ZOOM ERR MONITOR -> COLLECT, fe: 14
08/02/28 1122:01 1003 ZOOM ERR Send to COLLECT error: 201
08/02/28 1122:01 1003 ZOOM ERR Send to COLLECT error: 250

```

These error messages indicate that Expand communication problems exist between the Remote node and the Collection node.

ASAP Help

The ASAP CI has online help for all of the ASAP commands. For a summary of the help available for all of the ASAP commands:

```
+HELP ALL
```

This output is produced:

```

ASAP - Availability Stats and Performance - T0402V02.05 (15SEP2004)
ALL  ASAP commands are summarized below:
ACTION      Defines rules and lists actions.
APP         Reports on APPLICATION stats.
CIP         Reports on CIP stats.
CLEANUP     Invokes ASAP database cleanup.
COMM        Reports on COMM stats.
COMMIT      Instructs ASAP to commit all database changes.
CPU         Reports on CPU stats.
DB          Reports on database files when SET DBMAXDAYS is enabled.
DISK        Reports on DISK stats.
EDL         Performs EDL syntax check only, SET EDL compiles EDL.
EXIT        Stops the ASAP command interpreter.
FC          Fix Command.
FILE        Reports on FILE stats.
GOAL        Controls the objectives ASAP uses to produce alerts.
HELP        Displays explanation of specified ASAP command.
LH          Reports on Expand Line Handler stats.
LOG         Displays ASAP log event records.
MONITOR     Controls the set of objects monitored by ASAP.

```

New Features	Enter HELP NEW FEATURES for a new features history.
New User	Enter HELP NEW USER for getting started & product overview.
NODE	Reports on Expand end-to-end stats.
OBEY	Causes commands in OBEY file to be executed.
PAUSE	Suspends program until stop/abend/brk msg.
PB	Alias command for Reporting on busiest Process stats.
PROCESS	Reports on selected Processes.
PROCESSBUSY	Reports on busiest Process stats.
RANK	Controls the objectives ASAP uses to produce alerts.
RDF	Reports on Remote Database Facility (RDF) stats.
SET	Displays and/or alters option settings.
SHOW	Displays Entity information.
SHUTDOWN	Stops COLLECT and/or MONITOR in specified nodes.
SPOOLER	Reports on Spooler stats.
STARTUP	Starts COLLECT and/or MONITOR in specified nodes.
STATUS	Status COLLECT or MONITOR or ACTION server.
SWAP	Reports on KMSF stats.
SYNC	Manually requests MONITOR to sync specified nodes.
SYSTEM	Specifies default system name.
TAPE	Reports on TAPE stats and TAPE MOUNT REQUESTS.
TCP	Reports on the nine TCP/IP entities and the three Telserv entities.
TMF	Reports on Transaction Monitoring Facility (TMF) stats.
VOLUME	Changes/displays default volume name.
!	Delimits end of line comments.

To obtain detailed help on any command:

+HELP *command*

To obtain help on ASAP error numbers:

+HELP ERROR

All of the ASAP error messages are output to your terminal or workstation.

Section 4: Using Discrete Object Thresholds (DOTs)

DOTs let users set discrete objectives and recovery actions against specific attributes for any domain within an ASAP entity. DOTs also let users selectively monitor specific objects (domains), and provide an event generation service.

DOTs are implemented within ASAP by:

- Providing a database to store and retrieve domain names, objective values, and recovery actions
- A user command interface (ASAP CI)
- A recovery action service
- Automatic event generation service
- An internal service for calculating the ASAP OEM states and retrieving information from the database

Topic	Page
DOTs Overview	88
MONITOR Usage Considerations	95
GOAL and RANK Usage Considerations	96
EMS Event Considerations	96
System Entity Specifics	100

DOTs Overview

DOTs provide services to ASAP users in four ways:

- DOTs let you choose the objects to monitor with ASAP. Using the ASAP CI MONITOR command, you enter the list of objects (domains) in each subsystem to monitor with ASAP. For example, `MONITOR CPU 1` makes ASAP monitor processor 1. ASAP does not monitor any other processors unless they are included in the list to monitor.

Many ASAP components autoconfigure domains at startup if you do not enter any. For example, the Disk SGP automatically monitors all disk drives.

Other ASAP components autoconfigure a small set of domains. For example, the Process SGP automatically monitors only the ASAP Monitor process. When you add a domain to monitor to any subsystem, ASAP turns off autoconfiguration for that subsystem and only monitors the domains you specified. Adding a single domain, and turning it OFF in the objectives database, makes ASAP monitor no domains for the subsystem. For example, `MONITOR CPU 1, ADD, OFF` makes ASAP monitor no processors if it is the only CPU object specified in the database, or if all other CPU domains are also in the OFF state.

- DOTs lets you set simplex, escalating and Boolean thresholds on any attribute or attributes for any domain. Using the ASAP CI GOAL or RANK command, you set thresholds for subsystems and domains within subsystems on the attributes that are important to you. For example, `RANK CPU 0, BUSY < 90` sets a simplex objective on the Busy attribute for CPU 0 that it must be less than 90 percent busy. `RANK CPU 1, QUEUE < 1` sets an objective for CPU 1 that the queue length must be less than 1. `RANK CPU, CHIT > 25` sets a global objective for all processors that the cache hit rate must be greater than 25.

You set objectives using any of six operators for simplex objectives: greater than >; less than <; equal to =; not equal <>; greater than or equal to >=; and less than or equal to <=. A single attribute can have an objective for each operator.

For example, `RANK PROCESS $ABC, CPU > 2, CPU <> 5` sets two simplex objectives on the Process CPU attribute, process \$ABC must execute in a processor greater than 2 but not in processor 5. ASAP does not prevent you from setting objectives that are illogical and causes alerts at each interval. For example, `RANK PROCESS $ABC, CPU < 1, CPU > 1` cause an alert at each interval since the process cannot be in a processor less than 1 and a processor greater than 1.

Escalating objectives define up to three levels of less than or greater than thresholds for an attribute where each level has a different alert level and can have unique actions and events. For example, `GOAL CPU 1, BUSY < 40, BUSY << 60, BUSY <<< 80` defines 3 levels of alerting for the CPU Busy attribute for CPU 1. If the CPU is 45% busy the alert level will be a high alert; if the CPU is 65% busy the alert level will be a warning alert; and if the CPU is 85% busy the alert level will be a critical alert. The alert levels are predefined and are not controlled by the SET OBJECTIVESRANK setting.

Four operators are used to defined escalating goals: level 2 less than <<; level 3 less than <<<; level 2 greater than >>; and level 3 greater than >>>. When goals are defined using the level 2 or level 3 less than operators they are considered to be escalating goals and the value specified at each level should increase. When they are defined using level 2 or level 3 greater than operators they are considered to be descalating goals and the value specified should decrease at each level. Like simplex objectives, ASAP does not prevent you from setting illogical values for escalating goals.

Boolean objectives define multiple sets of simplex objectives in a single goal using AND and OR operators. A Boolean objective generates a critical alert for all attributes defined in the goal when it fails, and generates a single action and event if defined for the goal. For example, `GOAL CPU 1, BUSYGOAL = (BUSY < 80 AND QUEUE < 2)` defines a Boolean goal for the Busy and Queue length attributes for CPU 1. It states that the CPU must be less than 80% busy and the queue length must be less than 2. If either of these conditions is not true the goal will fail. This goal, `GOAL CPU 1, BUSYGOAL = (BUSY < 80 OR QUEUE < 2)`, states that the CPU must be 80% busy or the queue length must be less than 2. Both conditions must be false for the goal to fail.

The six simplex objective operators defined above can be used in Boolean goals; they can have multiple occurrences of AND and OR operators; and they can be nested using multiple levels of parenthesis. A total of eight attributes can be combined in various ways when using Boolean objectives. The evaluation order is left to right, starting with the innermost set of parentheses.

- DOTs lets you define recovery actions to be taken when objectives are not met. Using the ACTION parameter to the ASAP CI RANK command, a user can define commands to be sent to command interpreters such as SCF and TACL when specific objectives are not met. For example, the command `RANK CPU, STATUS ACTION "TACL RELOAD <#OBJECT>, PRIME" WAIT 5` causes ASAP to automatically reload a failed CPU, waiting 5 minutes between attempts. The token named <#OBJECT> is automatically replaced in the action string with the number of the failed processor. Other tokens that you can use in action strings for simplex and escalating goals:

<#ENTITY>	The entity name
<#OBJECT>	The NonStop™ OS or OSS object name
<#DOMAIN>	The domain name
<#D1>	Leftmost part of a multilevel domain name
<#D2>	2nd part of a multilevel domain name
<#D3>	3rd part of a multilevel domain name
<#D4>	4th part of a multilevel domain name
<#D5>	5th part of a multilevel domain name
<#ATTR>	The attribute name
<#OP>	The objective operator
<#STATE>	The current state of the attribute
<#LASTSTATE>	The last state of the attribute
<#VALUE>	The value of the attribute

<#GOAL> The objective value

Tokens are expanded when using Boolean goals. The <#ATTR>, <#OP>, <#VALUE> and <#GOAL> tokens can contain multiple tab-separated entries and new tokens are added as follows:

<#GNAME> The name of the Boolean goal

<#GOALVAL> The attribute, operator and value in a string with no spaces followed by the goal separated with the -|| character. There can be multiple tab-separated entries.

<#attrGOAL> A specific goal token for the attribute. For example, if the attribute is named BUSY the name of this token will be <#BUSYGOAL>. If the same attribute is used multiple times in the Boolean goal then the token name will be extended for each additional occurrence by using a numeric value immediately following the attribute name. For example the second occurrence will be named <#attr2GOAL>.

<#attrOP> A specific operator token for the attribute. For example, if the attribute is named BUSY the name of this token will be <#BUSYOP>. If the same attribute is used multiple times in the Boolean goal then the token name will be extended for each additional occurrence by using a numeric value immediately following the attribute name. For example the second occurrence will be named <#attr2OP>.

<#attrVALUE> A specific value token for the attribute. For example, if the attribute is named BUSY the name of this token will be <#BUSYVALUE>. If the same attribute is used multiple times in the Boolean goal then the token name will be extended for each additional occurrence by using a numeric value immediately following the attribute name. For example the second occurrence will be named <#attr2VALUE>.

Action strings can also be stored as predefined rules by using the ASAP CI ACTION command. ASAP creates an initial set of predefined rules automatically, including one called RELOAD that executes the TACL RELOAD command. The command RANK CPU, STATUS ACTION RELOAD WAIT 5 uses the predefined ASAP action rule instead of the TACL command string as shown in the previous example.

- DOTs lets you generate EMS events when objectives are not met. Using the ASAP CI GOAL or RANK command, you set objectives on attributes for any domain and include event generation keywords to make ASAP generate events.

For example, `RANK CPU 1, BUSY > 15 CRITICAL REPEAT` makes ASAP generate a critical EMS event when percent busy for processor 1 drops to 15% or below and keeps repeating that event for each interval where the problem persists.

DOTs require that you define entities using the ASAP Entity Definition Language (EDL) and that entity information is available for each caller of the DOTs API.

EDL lets you describe entity information required for the ASAP Client, Server, and Extension. Each product shares some information from EDL, and each has specific requirements for EDL information. ASAP Server must also keep non-EDL information in memory for each entity. For more information on EDL, see the *ASAP Client Manual*.

Specifying Domains and Objectives

The MONITOR command controls which objects, or domains, ASAP monitors. If no domains are specified, ASAP autoconfigures the domains it monitors. For example, if you SET DISK ON, all disks are monitored unless you use the MONITOR command to select specific disks.

Once a domain is monitored, use the GOAL or RANK command to set availability and performance objectives for that domain. When a ranked property upper or lower bound threshold is exceeded, the state of the ranked property is assigned an alert state. States in ASAP describe the status of a property such as up, down, warning, or critical. States can also indicate utilization such as low, medium, and high. ASAP displays ranked alert states in the ASAP Client, the EMS subsystem, or both.

To enable ranked property alerts in the ASAP Client, set State Determination to UseStateGraphState for each ranked property in the ASAP Client Attribute Property sheet. For more information, see the ASAP Client Manual.

To enable ranked property events in EMS, SET OBJECTIVESEVENT ON from the ASAP command interpreter before starting ASAP Server components. For more information, see SET OBJECTIVESEVENT option of the [SET Command](#).

The Objectives Database

The objectives database stores a list of domain names to monitor. If names do not exist in the database, a default set of objects are autoconfigured for monitoring.

The objectives database stores your objectives thresholds. ASAP produces states for attributes by comparing objectives, from simplex, escalating or Boolean goals defined in the database, to the value or values produced for attributes at each interval.

You define which entities and entity domains are stored in the objectives database file, which resides on each node managed by ASAP. You set the file name and location using the SET OBJECTIVESDB option of the [SET Command](#).

Host-based objectives constitute several types of objectives that you can define. You control which system or application entities and domains are monitored and define what the objective thresholds should be for an entity or a specific individual domain. For example, in a 16 CPU system, to configure ASAP to only monitor CPUs 0 through 5 and 12 through 15, use the ASAP MONITOR commands -Monitor CPU 0, ADDll, -Monitor CPU 1, ADDll, and so on, until all of the CPU entity domains are loaded in the ASAP objectives database.

The pre-defined system entities are:

- CIPCLIM
- CIPMONITOR
- CIPPROVIDER
- COMM
- CPU domains 0-15
- DISK
- EXPAND
- FILE
- HYBRID (Optional)
- PROCESS
- PROCESS BUSY domains 0-15
- RDF
- SPOOLER
- SWAP
- SYSTEM

- TAPE
- TCPARP
- TCPICMP
- TCPICMP6
- TCPIGMP
- TCPIP
- TCPIP6
- TCPLOOP
- TCPMONGQ
- TCPPort
- TCPProcess
- TCPQIO
- TCPRoute
- TCPRTE
- TCPSOCK
- TCPSubnet
- TCPUDP
- TelservProcess
- TelservService
- TelservWindow
- TMF

Once the monitored entities are loaded in the database, you can set individual user-defined objectives against entity domains by using the GOAL or RANK commands. For a complete list of configurable system entity objective properties, see [System Entity Specifics](#).

You configure objective thresholds for an entire entity or for a specific domain within an entity. For example, a BUSY attribute for the CPU entity could have BUSY < 50 and BUSY > 10, meaning the interval value reported for the attribute must be greater than 10 and less than 50. The commands for configuring this objective threshold are the GOAL and RANK commands. GOAL CPU, BUSY > 10, BUSY < 50. All CPU domains would have this objective threshold applied to them. For a detailed explanation of these commands, see the [GOAL Command](#) and the [RANK Command](#).

MONITOR Usage Considerations

This table shows the domains monitored by each ASAP if no objects are defined to monitor, and shows the default process names of the ASAP SGP processes and the database file name for each entity:

Entity	Default Process Name	Database Filename	Default Domains Monitored
APP	\$ZOOH	DBAPP	All application domains registered using the ASAP_REGISTER_API
CIP **	\$ZOOO	DBCIP	CIP CLIMs, Monitors and Providers
COMM	\$ZOOU	DBCOM	All configured communication line types 7, 11, 13, 53, 5158, 60, and 61
CPU	\$ZOO n ($n=0$ through 15)	DBCPU	All SYSGENed processors
DISK	\$ZOOJ	DBDSK	All physical disk drives
EXPAND *	\$ZOOK	DBEXP	All configured Expand line handlers
FILE	\$ZOOL	DBFIL	\$SYSTEM.SYSTEM.ASAP MON
NODE	\$ZOOG	DBNCP	Not applicable
PROCESS	\$ZOOT	DBPRO	\$asapidM, the ASAP Monitor process
PROCESSBUSY	\$ZOO n ($n=0$ through 15)	DBPCB	The five busiest processes in each processor
PROXY	\$ZOOO	DBAPP	All application domains from Linux systems registered

RDF	\$ZOON	DBRDF	using the Linux application API All RDF environments with RDF configuration files on \$SYSTEM
SPOOLER	\$ZOOO	DBSPL	\$SPLS
SWAP	\$ZOOA	DBSWP	All configured KMSF swap files
SYSTEM	\$ZOO _n (<i>n</i> =0 through 15)	DBSYS	Average utilization for all Up CPU's
TAPE	\$ZOOQ	DBTAP	All SYSGENed tape drives
TCP/IP **	\$ZOOV	DBTCP	All TCP/IP Processes, Routes and Subnets, and Telserv Processes
TMF	\$ZOOZ	DBTMF	All TMF components

* The Expand SGP is a multi-entity (Expand and ExpandIP) SGP. Use the Expand entity for setting monitoring of any Expand line. For use of the MONITOR command for Expand and ExpandIP, see [System Entity Specifics](#).

**The SGP is a multi-entity SGP. For more information, see [System Entity Specifics](#).

GOAL and RANK Usage Considerations

Use the GOAL and RANK commands to set simplex, escalating or Boolean objectives on any attribute defined in EDL that has an associated state (StatePair Yes). A simplex objective is a single goal set on an attribute; an escalating objective defines multiple alert levels for the same attribute; and a Boolean objective combines several goals into a single alert indicator.

For example, to set a global simplex objective for all CPUs to less than 90% busy with a queue length less than 2, then override it with a specific objective on CPU 2 that its queue length must be less than 1:

```
GOAL CPU, BUSY < 90, QUEUE < 2
```

```
GOAL CPU 2, QUEUE < 1
```

Note

GOAL and RANK are synonyms for the same command. You can interchange them at will.

To designate that Informative or Critical EMS events be generated when a simplex or escalating objective is not met, specify one of the keywords in the GOAL or RANK command. For example, to specify that a critical EMS event be generated if CPU 4 is busier than 95% and to repeat the event at each interval where the condition persists:

```
RANK CPU 4, BUSY <= 95 CRITICAL REPEAT
```

For ASAP to generate EMS events, you must turn EMS event generation on as described in [EMS Event Considerations](#). To load objective database changes into a running copy of ASAP, you must use the COMMIT command.

EMS Event Considerations

ASAP generates two classes of events:

- Normal status and problem events describing ASAP operation

These are events such as component startup messages or error messages when a component cannot obtain a system resource. They are written to EMS and the ASAPLOG log file, and are subject to event suppression if turned on within ASAP.

- Events generated by DOTs to describe the status of objects (domains) being monitored by ASAP

DOTs events are not written to the ASAPLOG log file or subject to event suppression within ASAP. Because the ASAP Client provides many features for monitoring the state of monitored domains, DOTs EMS event generation is turned off by default.

ASAP DOTs can generate EMS events for status changes and for failed objective comparisons:

- To turn on ASAP generation of down EMS events (ASAP event 4000) for failed domains and objectives, place the `SET ObjectivesEvent ON` command in your ASAPCONF file.
- To turn on the corresponding up event (event 4001) when the domain or objective recovers, place the `SET ObjectivesEventUp ON` command in your ASAPCONF file.

ASAP generates down events for a domain whenever the state for a goal configured to generate events is greater than or equal to the state specified in the `SET ObjectivesEventState` command. The default is CRITICAL, meaning that ASAP generates an EMS event when attribute states are critical or down. To generate events for warnings or lesser alert conditions, include the `SET ObjectivesEventState` command in your ASAPCONF file.

ASAP also lets you control the number of events written per domain. The `SET ObjectivesEventMax` value determines how many events ASAP writes per interval for a domain. The default is one, meaning ASAP writes one EMS event per domain. For simplex or escalating goals the event will be written for the goal set on the first attribute configured for events whose state is greater than or equal to the `ObjectivesEventState`. Since escalating goals can create High, Warning and Critical alerts `ObjectivesEventState` should be set to HIGH or lower to generate events and/or

actions at all levels. Any other attributes configured for events with states of equal or higher values are not alerted via EMS events. Boolean goals are processed after simplex and escalating goals, so no event will be written for Boolean goals if an event is written for a simplex or escalating goals. To have more than one event per domain per interval, place the `SET ObjectivesEventMax` command in your ASAPCONF file.

Other settings control whether events are displayed on the operator console. For more information, see the [SET Command](#).

To turn on down and up events for warnings and higher alerts, allowing two events per domain per interval, put these event commands in your ASAPCONF file:

```
SET ObjectivesEvent ON
SET ObjectivesEventUp ON
SET ObjectivesEventMax 2
SET ObjectivesEventState Warning
```

To create an informative EMS event when process \$Abc is not executing in CPU 1:

```
RANK PROCESS $ABC, CPU = 1 INFO
```

Status Change Events

ASAP DOTs automatically generates EMS events for Status changes in ASAP system entities when events are turned on. It uses the operational state—the associated state for the Status attribute—to determine whether to write an event. The operational state (OpState) is compared against the value defined in ObjectivesEventState. If OpState is greater than or equal to that value, an event is generated.

You cannot set objectives against the Status attribute. The operational status of a domain is always determined by the ASAP component that monitors the domain, and the associated operational state is set directly by ASAP. For example, if a monitored process fails, ASAP sets the Status to Down and sets OpState to 8 (the down state.) You cannot enter an objective to control that behavior.

Therefore, `RANK PROCESS, STATUS = "DOWN"` is an invalid command because objectives cannot be set on the Status attribute.

However, to control how ASAP generates events on Status changes, you can enter event control keywords using the Status attribute. By default, ASAP writes one critical EMS event when it finds a system object whose OpState passes the event generation test. If up events are on, it writes one corresponding up event when the object recovers.

To repeat the event at each interval, to have the event be an informative event, or to turn off events for the object altogether, use the RANK command:

- To change the event generated for all down processes to a repeating event:

RANK PROCESS, STATUS CRITICAL REPEAT

- To change the event generated for failures of \$Abc to an informative event:

RANK PROCESS \$ABC, STATUS INFO

- To turn off event generation for status changes to process \$Def:

RANK PROCESS \$DEF, STATUS NOEMS

By default, ASAP user-defined entities do not generate EMS events for Status changes. Because a single process can register hundreds of user domains with ASAP, failure of that process could generate hundreds of EMS messages. User-defined entities are created to describe your applications that use the ASAP Extension to register with ASAP. To turn on events for Status changes of user-defined entities:

- To generate a nonrepeating critical event when the Status changes to a bad state for any ATM domain:

RANK ATM, STATUS CRITICAL

- To generate a critical event and repeat it at each interval whenever \$Atm37 fails:

RANK ATM\CHICAGO\WEST\ATM37, STATUS CRITICAL REPEAT

The ASAP Extension provides an API that applications can use to set the Status and OpState for domains they register. Using the ASAPX API, an application gains direct control over when ASAP generates an event on behalf of that application.

Event Considerations Summary

- To turn on DOTs event generation, use the SET ObjectivesEvent ON and SET ObjectivesEventUp ON commands in your ASAPCONF file.
- To define when to generate events, set ObjectivesEventState, ObjectivesEventMax, and other parameters.
- To modify ASAP's automatic generation of Status change events for system objects, use GOAL, STATUS or RANK, STATUS.
- To define the additional attributes for which you want events generated, issue GOAL or RANK commands. For example, to generate a critical event if any CPU exceeds 89% busy, enter RANK CPU, BUSY < 90 CRITICAL.

System Entity Specifics

System entities that have one or more attributes with an associated StatePair attribute support DOTs. System entity attributes are defined in the file \$SYSTEM.SYSTEM.ASAP3SYS. Any modifications to this file might cause incorrect entity objectives records to be written to the ASAP objectives database.

To get help about which entity attributes can be ranked, use the SHOW command. For example:

```
SHOW FILE
```

```
File
  Nodename
  Sysnum
  Domain
  Op
  Date
  Time
  Valid
  Error
  Errorstate
  Full
  Fullstate
  Eof
  Eofstate
  Owner
  Ownerstate
  Rwp
  Rwpstate
  Open
  Openstate
  Tmf
  Tmfstate
  Files
  Filesstate
  Code
  Format
  Type
  Partitions
  Lastopen
  Lastmodif
```

An attribute defined with StatePair YES in the EDL definition is immediately followed by its associated state attribute. For example, ErrorState is the associated state for the Error attribute. When StatePair is true, you can set objectives on the attribute. ASAP places the state from the objectives comparison into the associated state attribute.

The STATUS field, the one marked StatelsOp in EDL, is a special field. It has an associated state, but objectives cannot be set against it. That operational state is controlled only by the SGP process monitoring that object.

These tables show all the attributes for each system entity that you can set objectives on:

CIPCLIM Entity-Supported Objective Attributes

Attribute	Description	Value
Error	Collection Error	Integer64
Present	Indicates whether an operational CLIM is present	Char 8
XStatus	Indicates whether the CLIM is connected to the ServerNet fabric	Char 8
YStatus	Indicates whether the CLIM is connected to the ServerNet fabric	Char 8
CIPsrvrs	Total number of socket servers	Integer16
Mode	Shows the operation mode of the CLIM	Char 8
ConnPts	Shows whether there are one or two ServerNet connections configured per fabric between CLIM and NSK host	Integer32
X2Status	Indicates whether the CLIM is connected to the ServerNet fabric	Char 8
Y2Status	Indicates whether the CLIM is connected to the ServerNet fabric	Char 8
Dataface	Total data interfaces	Integer32
DataIp	Total data IP addresses	Integer32
Mgmtface	Total mgmt interfaces	Integer32
MgmtIp	Total mgmt IP addresses	Integer32
Events	Number of syslog entries generated by the CLIMMON process	Integer32
Restarts	Number of times CLIMMON restarted the CLIMAGT and binary 32x processes	Integer32
Processes	Number of socket server processes	Integer32
AGTFails	Number of process failures	Integer32
SVRFails	Number of process failures	Integer32
AgtEvents	Number of event log entries	Integer32
AgtBfrDnys	Number of times a buffer for communication with NSK could not be obtained	Integer64
AgtApiErrs	Number of errors returned from the low-level ServerNet communication handlers	Integer64
AgtLnxErrs	Number of errors returned from CIP/Linux	Integer64
AgtBfrBytesCurr	Amount of memory currently used for communication with NSK	Integer64

AgtBfrBytesHigh	Highest amount of memory used for communication with NSK	Integer64
AgtRecvMsgs	Number of messages received from all NSK processors	Integer64
AgtSendMsgs	Number of messages sent to all NSK processors	Integer64
AgtConnCpusCurr	Current number of NSK processors connected	Integer32
AgtConnCpusHigh	Highest number of NSK processors connected	Integer32
AgtQueuedCmds	Number of SPI commands queued for processing	Integer32
AgtFailedCmds	Number of SPI commands that could not be completed	Integer32
AgtLastApiErr	Status code of the last IT-API error or zero if there has been no error	Integer32
AgtLastLnxErr	Errno value of the last CIP/Linux error or zero if there has been no error	Integer32
SvrEvents	Number of event log entries	Integer32
SvrBfrDnys	Number of times a buffer for communication with NSK could not be obtained	Integer64
SvrApiErrs	Number of errors returned from the low-level ServerNet communication handlers	Integer64
SvrLnxErrs	Number of errors returned from CIP/Linux	Integer64
SvrBfrBytesCurr	Amount of memory currently used for communication with NSK	Integer64
SvrBfrBytesHigh	Highest amount of memory used for communication with NSK	Integer64
SvrRecvMsgs	Number of messages received from all NSK processors	Integer64
SvrSendMsgs	Number of messages sent to all NSK processors	Integer64
SvrBytesSent	Number of data bytes sent to all NSK processors	Integer64
SvrBytesRcvd	Number of data bytes received from all NSK processors	Integer64
SvrConnCpusCurr	Current number of NSK processors connected	Integer32
SvrConnCpusHigh	Highest number of NSK processors connected	Integer32
SvrDfrdSnds	Number of times a send request could not be performed immediately	Integer32
SvrFailedCmds	Number of socket requests that could not be completed	Integer32
SvrLastApiErr	Status code of the last IT-API error or zero if there has been no error	Integer32
SvrLastLnxErr	Errno value of the last CIP/Linux error or zero if there has been no error	Integer32
SvrTcpLstnCurr	Number of TCP listen sockets	Integer32
SvrTcpLstnHigh	Highest number of TCP listen sockets	Integer32
SvrUdpScktsCurr	Number of UDP listen sockets	Integer32
SvrUdpScktsHigh	Highest number of UDP listen sockets	Integer32
SvrTcpConnsCurr	Number of TCP connections	Integer32
SvrTcpConnsHigh	Highest number of TCP connections	Integer32

CIPMONITOR Entity-Supported Objective Attributes

Attribute	Description	Value
Error	Collection Error	Integer64
MemUsed	Percent heap space used by the CIPMON process	Integer
Cpu	CIPMON CPU	Integer
Pin	CIPMON pin	Integer
Pri	CIPMON priority	Integer
QioPoolCur	Size of the QIO pool	Integer64
QioPoolLmt	Max size of OIO pool CIPMON can allocate or 0 for no limit	Integer64
RcvScktReqs	Number of receive socket requests across all CLIMs	Integer64
RcvScktReqErr	Number of receive socket requests that returned an error	Integer64
SndScktReqs	Number of send socket requests across all CLIMs	Integer64
SndScktReqErr	Number of send socket requests that returned an error	Integer64
BytesRcvd	Number of data bytes received from all CLIMs	Integer64
BytesSent	Number of data bytes sent to all CLIMs	Integer64
InConns	Number of connection requests received from all CLIMs	Integer32
OutConns	Number of connection requests sent to all CLIMs	Integer32
TcpSktCurr	Number of TCP listen sockets	Integer32
TcpSktHigh	Highest number of TCP listen sockets	Integer32
UdpScktsCurr	Number of UDP listen sockets	Integer32
UdpcSktsHigh	Highest number of UDP listen sockets	Integer32
TcpConnsCurr	Number of TCP connections	Integer32
TcpConnsHigh	Highest number of TCP connections	Integer32
Size128	Number of socket send requests between 1 and 128 bytes across all CLIMs	Integer32
Size256	Number of socket send requests between 129 and 256 bytes across all CLIMs	Integer32
Size512	Number of socket send requests between 257 and 512 bytes across all CLIMs	Integer32
Size1024	Number of socket send requests between 513 and 1024 bytes across all CLIMs	Integer32
Size2048	Number of socket send requests between 1025 and 2048 bytes across all CLIMs	Integer32
Size4096	Number of socket send requests between 2049 and 4096 bytes across all CLIMs	Integer32
Size8192	Number of socket send requests between 4097 and 8192 bytes across all CLIMs	Integer32
Size12288	Number of socket send requests between 8193 and 12288 bytes across all CLIMs	Integer32

Size16384	Number of socket send requests between 12289 and 16384 bytes across all CLIMs	Integer32
Size32768	Number of socket send requests between 16385 and 32768 bytes across all CLIMs	Integer32
Size57344	Number of socket send requests greater than 32768 bytes across all CLIMs	Integer32
ConnClimCurr	Number of CLIMs connected	Integer32
ConnClimHigh	Highest number of CLIMs connected	Integer32
RejectedConns	Number of attempts to connect to a CLIM that were rejected	Integer32
DeferredSends	Number of times a send request could not be performed immediately	Integer32
MsgsSent	Number of messages sent to all CLIMs	Integer64
MsgsRecv	Number of messages received from all CLIMs	Integer64

CIPPROVIDER Entity-Supported Objective Attributes

Attribute	Description	Value
Error	Collection Error	Integer64
ProvType	Type of Provider	Char 8
IpFamily	Network mode of the Provider	Char 6
TpStatus	Status of Transport Provider process associated with the Provider	Char 8

COMM Entity-Supported Objective Attributes

Attribute	Description	Value
CPU	Primary CPU Comm Line configured to	Integer
BCPU	Backup CPU Comm Line configured to	Integer
Requests	IO Requests per second	Integer32
Busy	Percent time line busy reading and writing	Integer
ReadBusy	Percent Time Line Busy Reading	Integer
WriteBusy	Percent Time Line Busy Writing	Integer
InKB	Kilobytes read per second	Integer64
OutKB	Kilobytes written per second	Integer64
InDataKB	Data kilobytes read per second	Integer64
OutDataKB	Data kilobytes written per second	Integer64
Rate	Total of reads, writes per second	Integer64
Reads	Number of read operations per second	Integer32
Writes	Number of write operations per second	Integer32
Retries	Total Number of I/O retries	Integer32

Transactions	Number of terminal transactions all subdevices	Integer32
RespTime	Response time per terminal transaction	Integer64

CPU Entity-Supported Objective Attributes

Attribute	Description	Value
BUSY	Percent busy	Integer
QUEUE	Queue length	Integer
DISP	Dispatch rate	Integer
DISK	Disk I/O rate per second	Integer
CHIT	Disk cache hit rate per second	Integer
SWAP	Memory manager page faults per second	Integer
MEMQ	Average memory manager queue length	Integer
PGS	Percent of total physical memory pages locked	Integer
PCB	Percent usage low process control blocks (pins 0 through 254)	Integer
PCBX	Percent usage of high process control blocks pins above 255)	Integer
TLE	Percent usage of time list elements	Integer
SEGS	Percent of maximum virtual memory segments used	Integer
BrkPnt	Debug break points in use	Integer
Blades	Number of Integrity Blades that are Up	Integer
MemSize	Physical memory size	Integer32
MemSwap	Swappable physical memory	Integer32
MemFree	Free physical memory	Integer32
MemLock	Locked physical memory	Integer32
MemLockMax	Maximum memory that can be locked in physical memory	Integer32
MemLockHigh	Highest amount of locked memory since CPU was loaded	Integer32
IPUs	Number of IPU's for the logical CPU	Integer

DISK Entity-Supported Objective Attributes

Attribute	Description	Value
P *	Primary disk path in use, value = P or B	Char 1 wide
M **	Mirror disk path in use, value = P or B	Char 1 wide
CPU	Primary CPU number of the Disk process for this volume	Integer
FULL	Percentage of disk space currently used	Integer

REQUESTS	Number of disk I/O requests per second	Integer32
BUSY	Percent time disk busy reading, writing, and seeking	Integer
WRITE	Percent time disk busy writing	Integer
READ	Percent time disk busy reading	Integer
CHIT	Total cache hits per second for all block sizes	Integer
SWAP	Number of disk page swaps per second	Integer
QLEN	Disk process receive queue length	Integer64 (implied decimal)
INKB	Input kilobytes per second	Integer32
OUTKB	Output kilobytes per second	Integer32
RATE	Total of DP2 reads, writes, and seeks per second	Integer
READS	Number of DP2 read operations per second	Integer
WRITES	Number of DP2 write operations per second	Integer
BSPLTS	Total volume block splits for all block sizes	Integer
REQBLK	Number of disk requests blocked per second	Integer
PCBX	Percent usage of high process control blocks pins above 255)	Integer
TLE	Percent usage of time list elements	Integer
SEGS	Percent of maximum virtual memory segments used	Integer
Fragment	Largest disk fragment available in megabytes	Integer32

* To use -P path rank disk P = P; to use -B path rank disk M = B.

** To use -M path rank disk M = P; to use -MB path rank disk M = B.

EXPAND Entity-Supported Objective Attributes

Attribute	Description	Value
POOL	Percent utilization of Expand line handler I/O pool	Integer
PFAIL	Number of pool failures during the sample interval	Integer32
NoBuf	Number of no buffer failures	Integer32
BCC	Number of BCC errors during the sample interval	Integer32
NakSnt	Number of Level-4 negative acks sent	Integer32
NakRvd	Number of Level-4 negative acks received	Integer32
PktSnt	Number of Level-4 packets sent	Integer32
PktRvd	Number of Level-4 packets received	Integer32
PThruSnt	Number of Level-4 PassThru packets sent	Integer32
PThruRvd	Number of Level-4 PassThru packets received	Integer32
LnkSnt	Number of requests sent	Integer32
LnkRvd	Number of requests received	Integer32

CPU	Primary Cpu number of the Expand Line Handler Process for this line	Integer
-----	---	---------

EXPANDIP Entity-Supported Objective Attributes

Attribute	Description	Value
DFramSnt	Number of IP/ATM data frames sent	Integer32
DFramRvd	Number of IP/ATM data frames received	Integer32
DFByteSnt	Data kilobytes sent per second	Integer32
DFByteRvd	Data kilobytes received per second	Integer32
CnctCmdSnt	Number of IP/ATM connect commands sent	Integer32
CnctCmdRvd	Number of IP/ATM connect commands received	Integer32
CnctRspSnt	Number of IP/ATM connect responses sent	Integer32
CnctRspRvd	Number of IP/ATM connect responses received	Integer32
QryCmdSnt	Number of query commands sent	Integer32
QryCmdRvd	Number of query commands received	Integer32
QryRspSnt	Number of query responses sent	Integer32
QryRspRvd	Number of query responses received	Integer32
MemLow	Number of times memory-low indication given by QIO	Integer32
InvFrn	Number of IP/ATM invalid frames received	Integer32
DrpFrm	Number of IP/ATM frames dropped	Integer32
InvAdr	Number of IP/ATM frames received from an unexpected system	Integer32

FILE Entity-Supported Objective Attributes

Attribute	Description	Value
Error	Error number indicating status of data	Integer64
Full	Percentage full	Integer
Eof	Size in bytes	Integer64
Owner	Group, User	Character 7 wide
RWEP	Read, Write, Execute, Purge security (Safe = Safeguard)	Character 4 wide
Open	Yes = Open; No = Not open	Character 3 wide
TMF	Yes = Audited by TMF; No = Not audited	Character 3 wide
Files	Number of files in subvolume	Integer32
Permission	OSS permissions string	Character 10 wide

Progid	Progid indicator (yes/no)	Character 3 wide
License	License indicator (yes/no)	Character 3 wide

HYBRID Entity-Supported Objective Attributes

Attribute	Description	Value
Error	Collection Error	Integer64
Nak	Negative Acknowledgement Count	Integer64
DataGrams	Datagram arrival rate	Real64
Samples	Sample datagram arrival rate	Real64
RegCnt	Count of ASAP_REGISTER_ calls in this interval	Real64
UpdCnt	Rate (per second) of ASAP_UPDATELIST_ calls for this interval	Real64
RmvCnt	Count of ASAP_REMOVE_ calls in this interval	Real64
ErrCnt	Count of API errors in this interval	Real64
NetTime	Average network turnaround time during interval, in seconds	Real64
SysCnt	Count of systems heard from in this interval	Real64
OpCnt	Count of ASAP_OPSTATE_ calls in this interval	Real64
CtrlCnt	Count of ASAP_CONTROL_ calls in this interval	Real64

PROCESS Entity-Supported Objective Attributes

Attribute	Description	Value
Error	Collection error	Integer64
Cpu	Process primary CPU	Integer
Pri	Primary process priority	Integer
BCpu	Process backup CPU	Integer
Busy	Process Busy	Integer64 (implied decimal)
CBusy	CPU Busy for non-Blades systems, or Average CPU busy across all IPUs on Blades systems	Integer64 (implied decimal)
Affnty	IPU Affinity - Unknown (None), Hard, Soft, Group	Char 8
IPU	IPU of the Primary Process	Integer16
BIPU	IPU of the Backup Process (if NonStop™)	Integer16
PState	Unalloc, Start, Run, Suspnd, Dbg brk, Dbg trap, Dbg req, Insp	Character 8 wide

WState	Lsig/Lpipe(status), Pon(cpu pwr), Iopon (I/O pwr), Intr	Character 8 wide
MPages	Memory pages in use	Integer64 (implied decimal)
MSent *	Messages sent per second	Integer64 (implied decimal)
MRecvd *	Messages received per second	Integer64 (implied decimal)
Qlen *	Receive queue length	Integer32
MQlen *	Maximum receive queue length	Integer32
PFaults *	Page faults per second	Integer64 (implied decimal)
Count	Count of members in aggregate	Integer32
ACount	Number of domains/attributes alerting	Integer32

* Requires Measure

PROCESSBUSY Entity-Supported Objective Attributes

Attribute	Description	Value
Busy	IPU busy for Blades systems, CPU busy for others	Integer
CBusy	Average busy across IPU's on Blades systems	Integer
RcvQ	Process receive queue length integrated	Integer
RcvMsg	Messages received per second	Integer
SndMsg	Messages sent per second	Integer
Pages	Number of memory manager pages allocated to process	Integer
LCB	Number of LCBs in use by the process	Integer

RDF Entity-Supported Objective Attributes

Attribute	Description	Value
Error	Error number indicating status of data	Integer
RTDSEcs	Relative time delay in seconds	Integer32
RTDTime	Relative time delay in <i>hh:mm:ss</i>	Integer32
PCpu	Primary CPU	Integer
BCpu	Backup CPU	Integer
Priority	Execution priority	Integer

SPOOLER Entity-Supported Objective Attributes

Attribute	Description	Value
Error	Last error encountered	Integer32
PCPU	CPU of primary process	Integer
BCPU	CPU of backup process	Integer
NumJobs	Number of jobs defined for this entity	Integer32
BatchJobs	Number of batch jobs defined for this entity	Integer32
OpenJobs	Number of jobs not fully defined	Integer32
PrintJobs	Number of jobs currently printing	Integer32
HoldJobs	Number of jobs completed and on hold	Integer32
NumDevQ	Number of device queue nodes	Integer32
NumLocs	Number of locations defined	Integer
NumFonts	Number of font jobs defined	Integer
FullPCT	Percentage of collector units currently used	Integer

SWAP Entity-Supported Objective Attributes

Attribute	Description	Value
Files	Number of KMSF files for the CPU	Integer
SwapTotal	Total virtual Swap space allocated in all KMSF files	Integer32
Reserved	Total Reserved space for all swap files	Integer
Reservations	Sum of outstanding KMSF reservations for all segments	Integer32
Available	Total Available space for all swap files	Integer32
Full	Percentage full for all swap files	Integer

Avail	Percentage available for all swap files	Integer
Creates	Reservations made for process creates during the sample interval	Integer32
CreateFails	Failed reservations for process creates during the sample interval	Integer32
Releases	Reservations released during the sample interval	Integer32
ReleaseFails	Failed reservations released during the sample interval	Integer32
Resizes	Successful reservation resizes during the sample interval	Integer32
ResizeFails	Failed reservation resizes during the sample interval	Integer32
Allocates	Successful allocations made in the swap files during the sample interval	Integer32
AllocateFails	Failed allocations in the swap files during the sample interval	Integer32
FreesOk	Successful freed allocations during the sample interval	Integer32
FreesFails	Failed freed allocations during the sample interval	Integer32

SYSTEM Entity-Supported Objective Attributes

Attribute	Description	Value
BUSY	Percent busy	Integer
QUEUE	Queue length	Integer
DISP	Dispatch rate	Integer
DISK	Disk I/O rate per second	Integer
CHIT	Disk cache hit rate per second	Integer
SWAP	Memory manager page faults per second	Integer
MEMQ	Average memory manager queue length	Integer
PGS	Percent of total physical memory pages locked	Integer
PCB	Percent usage of low process control blocks (pins 0 through 254)	Integer
PCBX	Percent usage of high process control blocks (pins above 255)	Integer
TLE	Percent usage of time list elements	Integer
SEGS	Percent of maximum virtual memory segments used	Integer
BrkPnt	Debug break points in use	Integer
Blades	Number of Integrity Blades that are Up	Integer
MemSize	Physical memory size	Integer32

MemSwap	Swappable physical memory	Integer32
MemFree	Free physical memory	Integer32
MemLock	Locked physical memory	Integer32
MemLockMax	Maximum memory that can be locked in physical memory	Integer32
MemLockHigh	Highest amount of locked memory since CPU was loaded	Integer32

TAPE Entity-Supported Objective Attributes

Attribute	Description	Value
PCPU	CPU of the backup tape device process	Integer
BCPU	CPU of the backup tape device process	Integer
MountReq	Number mount requests outstanding for this tape drive or node	Integer
MountTime	Amount of time since MountReq > 0	Integer
TapeStatus	Status of the tape currently loaded in the tape drive	Character 8 wide
TapeLabel	Label type of the tape	Character 7 wide

TCP/IP and Telserv Entities-Supported Objective Attributes

All of these entities/attributes are not supported on all versions of TCP/IP. If an entity/attribute is not supported it will always be reported with a zero value.

Attribute	Description	Value
ENTITY TcpArp	Address Resolution Protocol	
Error	Collection Error	Integer64
SntArpreq	ARP requests sent	Integer32
RcvArpreq	ARP requests received	Integer32
SntArprsp	ARP replies sent	Integer32
RcvArprsp	ARP replies received	Integer32
SntInvreq	Inverse ARP requests sent	Integer32
RcvInvreq	Inverse ARP requests received	Integer32
SntInvrsp	Inverse ARP replies sent	Integer32
RcvInvrsp	Inverse ARP replies received	Integer32
SntNaks	Number of ARP NAKs sent	Integer32

RcvNaks	Number of ARP NAKs received	Integer32
SntProbes	Number of ARP probes sent	Integer32
RcvProbes	Number of ARP probes received	Integer32

ENTITY Tcplcmp Internet Control Message Protocol

Error	Collection Error	Integer64
Err	Number of ICMP error packets that were generated	Integer32
OldShort	Number of short IP/ICMP packets	Integer32
Oldcmp	Number of bad ICMP Packets	Integer32
OutEchoRply	Number of Echo message output replies	Integer32
OutDstUnrch	Number of Destination Unreachable messages output	Integer32
OutSrcQnch	Number of Source Quench messages output	Integer32
OutRedirect	Number of Redirect messages output	Integer32
OutEcho	Number of Echo messages output	Integer32
OutTimeExcd	Number of Time Exceeded messages output	Integer32
OutParamPrb	Number of Parameter Problem messages output	Integer32
OutTimeStmp	Number of Timestamp messages output	Integer32
OutTimeSRpl	Number of Timestamp Reply messages output	Integer32
OutInfoReq	Number of Information Request messages output	Integer32
OutInfoRpl	Number of Information Reply messages output	Integer32
RteSolicit	Number of router solicitation messages	Integer32
Badcode	Number of ICMP packets that had invalid ICMP codes	Integer32
TooShort	Number of packets that were too short	Integer32
Xsum	Number of packets received with bad checksum values	Integer32
BadLen	Number of packets that were shorter than the length specified in the ICMP header	Integer32
Reflect	Number of ICMP Reflect packets that were received	Integer32
InEchoRply	Number of Echo message input replies	Integer32
InDstUnrch	Number of Destination Unreachable messages input	Integer32
InSrcQnch	Number of Source Quench messages input	Integer32
InRedirect	Number of Redirect messages input	Integer32

InEcho	Number of Echo messages input	Integer32
InTimeExcd	Number of Time Exceeded messages input	Integer32
InParamPrb	Number of Parameter Problem messages input	Integer32
InTimeStmp	Number of Timestamp messages input	Integer32
InTimeSRpl	Number of Timestamp Reply messages input	Integer32
InInfoReq	Number of Information Request messages input	Integer32
InInfoRpl	Number of Information Reply messages input	Integer32
RteAdvert	Number of router advertisement messages	Integer32
BadRapSubcode	Number of IRDP messages with bad ICMP subcodes	Integer32
BadAddrlist	Number of IRDP messages with a bad address list	Integer32
BadWpa	Number of IRDP messages with an incorrect address length	Integer32
RdiscNrecorded	Number of valid routes discovered by IRDP messages that have been entered in TCPIP's route table	Integer32

ENTITY Tcplcmp6 Internet Control Message Protocol v6

Error	Collection Error	Integer64
TxError	ICMPv6 error packets sent	Integer32
TxErrorDrops	Error packets not sent	Integer32
TxNobufs	Error packets not sent due to no buffers	Integer32
TxRlimitDrops	Error packets not sent due to rate limit	Integer32
RxTotal	ICMPv6 packets received	Integer32
RxInfo	ICMPv6 informational class packets received	Integer32
RxError	ICMPv6 error class packets received	Integer32
RxTooshort	Short packets received	Integer32
RxChecksum	Packets received with bad checksum	Integer32
RxNotifications	Notification packets received	Integer32
RxTorawip6	Packets passed to possible ICMPv6 listeners	Integer32
RcvQueries	Queries received	Integer32
RcvBadqueries	Bad queries received	Integer32
RcvReports	Reports received	Integer32
RcvBadReports	Bad reports received	Integer32
RcvOurReports	Our reports received	Integer32
InDstUnreach	Destination unreachable messages received	Integer32

InPackTooBig	Packet too big messages received	Integer32
InTimeExceed	Time exceeded messages received	Integer32
InParamProb	Parameter problem messages received	Integer32
InEchoReq	Echo request messages received	Integer32
InEchoReply	Echo reply messages received	Integer32
InMldQuery	Multicast listener discovery queries received	Integer32
InMldReport	Multicast listener discovery reports received that match our listeners	Integer32
InNdRtrAdvert	Router advertisement packets received	Integer32
InNdNbrSol	Neighbor solicitation packets received	Integer32
InNdNbrAdv	Neighbor advertisement packets received	Integer32
InNdRdirect	Redirect packets received	Integer32
OutDstUnrech	Destination unreachable messages sent	Integer32
OutPackTooBg	Packet too big messages sent	Integer32
OutTimeExced	Time exceeded messages sent	Integer32
OutParamProb	Parameter problem messages sent	Integer32
OutEchoReq	Echo request messages sent	Integer32
OutEchoReply	Echo reply messages sent	Integer32
OutMldReport	Multicast listener discovery reports sent	Integer32
OutMldReduct	Multicast listener discovery done sent	Integer32
OutNdRtrSol	Router solicitation packets sent	Integer32
OutNdNbrSol	Neighbor solicitation packets sent	Integer32
OutNdNbrAdv	Neighbor advertisement packets sent	Integer32
NdBadSolicit	Bad neighbor solicitation packets	Integer32
NdBadAdvert	Bad neighbor advertisement packets	Integer32
NdBadOptions	ND6 packets with bad options	Integer32
NdBadChelkp	ND6 cache lookups	Integer32
NdBadCheupd	ND6 cache updates	Integer32
NdBadRdirect	Bad redirect messages received	Integer32
RtrBadSol	Bad ND6 router solicitation messages received	Integer32
RtrBadAdv	Bad ND6 router advertisement messages received	Integer32
ENTITY Tcplgmp	Internet Group Management Protocol	
Error	Collection Error	Integer64
RcvTotal	Total number of IGMP packets received	Integer32

RcvTooshort	Number of packets received that were too short	Integer32
RcvBadsum	Number of packets received that had an incorrect checksum	Integer32
RcvQueries	Number of IGMP Query packets received	Integer32
RcvBadQueries	Query packets received with IP destination address not equal to the all hosts group	Integer32
RcvReport	Number of IGMP membership reports received	Integer32
RcvBadReports	Number of bad membership reports received	Integer32
RcvOurReports	Number of IGMP membership reports received for groups we belong too	Integer32
SndReports	Number of IGMP membership reports sent	Integer32

ENTITY Tcplp Internet Protocol

Error	Collection Error	Integer64
Total	Total number of input packets received	Integer32
BadXsum	Packets with bad checksum	Integer32
TooShort	Packets that were too short	Integer32
TooSmall	Packets without enough data	Integer32
BadHdrLen	Packets with bad header length	Integer32
BadLen	Packets with invalid length	Integer32
Frag	Fragments received	Integer32
FragDrop	Fragments dropped	Integer32
FragTimeout	Fragments whose TTL expired	Integer32
Fwd	Packets forwarded	Integer32
CantFwd	Packets that could not be forwarded	Integer32
RedirSent	ICMP redirects sent	Integer32
OutPkts	Total output packets	Integer32
NoProto	Unknown or unsupported protocol	Integer32
Delivered	Datagrams delivered to upperlevel	Integer32
LocalOut	Total ip packets generated here	Integer32
ODropped	Lost packets due to nobufs, etc	Integer32
Reassembled	Total packets reassembled OK	Integer32
Fragmented	Datagrams successfully fragmented	Integer32
OFragmented	Output fragments created	Integer32
CantFrag	Don't fragment flag was set, etc	Integer32
BadOptions	Error in ip option processing	Integer32

NoRoute	Packets discarded due to no route	Interger32
BadVers	Bad ip version	Interger32
RawOut	Total raw ip packets generated	Interger32
BadSrcIntf	Incorrect source interface or no route	Interger32
FragOverflow	Fragments that exceeded limit	Interger32

ENTITY Tcplp6 Internet Protocol v6

Error	Collection Error	Integer64
RxTotal	Total packets received	Interger32
RxIdleDiscards	Packets received over a subnet not configured for IPv6	Interger32
RxTooSmall	Packet length less then the size of an IPv6 header	Interger32
RxTooShort	Packets with lengths shorter than specified in their header	Interger32
RxBadVers	Version number not equal to 6	Interger32
RxErrors	Received packets resulting in the sending of an ICMPv6 error message	Interger32
RxLocal	Packets with one of our addresses received	Interger32
RxNonlocal	Packets received with an address that does not match one of ours	Interger32
RxMulticast	Multicast packets received	Interger32
RxBadAligns	Packets adjusted because the IPv6 header was not aligned correctly	Interger32
RxNoProto	Packets received and dropped with an unknown protocol	Interger32
RxDelivered	Packets delivered to upper level protocols	Interger32
RxDiscarded	Packets discarded	Interger32
RxForwarded	Source-routed packets forwarded	Interger32
RxTentative	Packets received on a tentative address	Interger32
TxLocal	Total packets output	Interger32
TxFragmented	Packets that were fragmented	Interger32
TxQuotaExceeded	Transmit quota exceeded	Interger32
TxMulticast	Multicast packets	Interger32
TxForward	Packets forwarded	Interger32
TxAttempts	Packet send attempts	Interger32
TxFailures	Packet send failures	Interger32
TxLoopedMcasts	Multicast packets looped back	Interger32

TrlTrFailure	Trl transmit failure	Interger32
EncAllocFailed	Enc allocation failure	Interger32
RipNoProto	No matching application for packet	Interger32
RipNoSpace	No space in socket for packet	Interger32
RipDelivered	Raw input packets delivered	Interger32
RipAllocFailed	Raw input allocation failures	Interger32
RipTooShort	Raw input packets too short	Interger32
RipBadcksum	Raw input packets with bad checksum	Interger32
FragPackets	Packets fragmented	Interger32
FragFragments	Number of fragments built	Interger32
FragNoFragPart	Fragment not found	Interger32
FragBadHopByHop	Packets with hop by hop header not after IPv6 header	Interger32
FragUfpBadAlign	Packets with unfragmented part not quad word aligned	Interger32
FragUfpTooBig	Transmit packets with un-fragmented part too big	Interger32
FragHdrTooBig	Packets with fragment header too big	Interger32
FragJumbo	Packets with jumbo payload option	Interger32
FragAllocFailed	Fragment failures due to no memory	Interger32
ReasFrgsRcvd	Fragments received	Interger32
ReasPktsReass	Packets successfully reassembled	Interger32
ReasTimeout	Packet reassembly timeouts	Interger32
ReasCopy	Unfragmented part of packets copied for reassembly	Interger32
ReasUfpTooBig	Unfragmented part of received packet too big	Interger32
ReasBadFragLen	More fragment flag set and packet length not a multiple of 8	Interger32
ReasPyldTooBig	Reassembled packets payload too big	Interger32
ReasOverlap	Partial fragment overlaps detected during reassembly	Interger32
ReasDupfrag	Duplicate fragments received, packets dropped	Interger32
ReasJumbo	Received fragments with jumbo payload option	Interger32
ReasAllocFailed	Allocation failures during packet reassembly	Interger32
FwdSuccess	Total packets forwarded	Interger32
FwdBadDst	Destination not forwardable	Interger32
FwdBadSrc	Source not forwardable	Interger32
FwdEnoBufs	No buffers available	Interger32
FwdEmsgSize	Packet too big	Interger32

FwdEother	Address unreachable	Integer32
-----------	---------------------	-----------

ENTITY TcpLoop Loopback

Error	Collection Error	Integer64
WhoTcpSent	Loopback TCP sent	Integer32
WhoTcpRcvd	Loopback TCP received	Integer32
WhoUdpSent	Loopback UDP sent	Integer32
WhoUdpRcvd	Loopback UDP received	Integer32
HaveTcpSent	TCP found sent	Integer32
HaveTcpRcvd	TCP found received	Integer32
HaveUdpSent	UDP found sent	Integer32
HaveUdpRcvd	UDP found received	Integer32
PacketSent	Packets sent	Integer32
PacketRcvd	Packets received	Integer32
UpdateSent	Update port sent	Integer32
UpdateRcvd	Update port sent	Integer32

ENTITY TcpMongq Monitor Global Queue

Error	Collection Error	Integer64
AllMonsSent	Requests sent to all MONs	Integer32
SomeMonsSent	Requests sent to one MON	Integer32
AllButMeSent	Requests sent to all MONs but me	Integer32
DpasRcvd	DPAS received	Integer32
MigrRcvd	Socket migrations received	Integer32
LoopbkRcvd	Loopback received	Integer32
IpfragRcvd	IP fragments received	Integer32
UnkRcvd	Unknown format received	Integer32
BadFrmtRcvd	Bad format received	Integer32

ENTITY TcpPort Port (Socket)

Error	Collection Error	Integer64
Opener	Process the opened the TCP/IP Port	Character 8 wide
CAID	Opener creator access ID	USERID

PAID	Opener process access ID	USERID
PFName	Program File Name	Character 34 wide
Cpu	Opener CPU	Integer16
Pin	Opener PIN	Integer16
Fnum	Primary File Number	Integer16
BCpu	Opener backup CPU if any	Integer16
BPin	Opener backup PIN if any	Integer16
BFnum	Backup File Number if any	Integer16
Proto	Protocol	Character 6 wide
LPort	Local port	Integer16
LAddr	Local TCP/IP address	IPADDR
FPort	Foreign port	Integer16
FAddr	Foreign TCP/IP address	IPADDR
SendQ	Bytes in send queue of socket	Integer16
RecvQ	Bytes in receive queue of socket	Integer16
Family	Protocol family	Integer16
ZLocAddr6	Local TCP/IP v6 address	IPADDR6
ZForgnAddr6	Foreign TCP/IP v6 address	IPADDR6

ENTITY TCP/IP Process/Monitor
TcpProcess

Error	Collection Error	Integer64
Sockets	The total number of sockets in use	Integer32
SockSPI	Number of sockets open using SPI protocol	Integer32
SockTCP	Number of sockets open using TCP protocol	Integer32
SockUDP	Number of sockets open using UDP protocol	Integer32
SockUnkn	Number of sockets open using an unknown protocol	Integer32
SockLstn	The open sockets currently in a listening state	Integer32
Connattempt	Number of connections initiated	Integer32
Accepts	Number of connections accepted	Integer32
Connects	Number of connections established	Integer32
Drops	Number of connections dropped	Integer32
Conndrops	Number of embryonic connections dropped	Integer32
Closed	Number of connections closed (includes drops)	Integer32
Segstimed	Number segments where TCP tried to get round	Integer32

	trip time	
Rttupdated	Number times TCP succeeded getting round trip time	Integer32
Delack	Number of delayed ACKs sent	Integer32
TimeoutDrop	Number of connections dropped in retransmit timeout	Integer32
RexmtTimeo	Number of retransmit timeouts	Integer32
PersistTimeo	Number of persist timeouts	Integer32
KeepTimeo	Number of keepalive timeouts	Integer32
KeepProbe	Number of keepalive probes sent	Integer32
KeepDrops	Number of connections dropped in keepalive	Integer32
SndTotal	Total packets sent	Integer32
SndPack	Data packets sent	Integer32
SndByte	Data bytes sent	Integer32
SndRexmitPack	Data packets retransmitted	Integer32
SndRexmitByte	Data bytes retransmitted	Integer32
SndRexmitFast	Fast retransmits	Integer32
SndAcks	ACK-only packets sent	Integer32
SndProbe	Window probes sent	Integer32
SndUrg	Packets sent with URG only	Integer32
SndWinup	Window update-only packets sent	Integer32
SndCtrl	Control (SYN/FIN/RST) packets sent	Integer32
BadHdrLen	Header Length Invalid (V4 Only)	Integer32
RcvTotal	Total packets received	Integer32
RcvPack	Packets received in sequence	Integer32
RcvByte	Bytes received in sequence	Integer32
RcvBadSum	Number of packets with bad checksum	Integer32
RcvBadOff	Number of packets with bad offset	Integer32
RcvShort	Packets received too short	Integer32
BadSeqs	Packets with bad sequence number - HPE specific	Integer32
UnAck	Packets unACKed - HPE specific	Integer32
Timeout	Packets timeout - HPE specific	Integer32
NoPort	Packets with no-port - HPE specific	Integer32
RcvUrg	Packets received with URG set - HPE specific	Integer32
RcvDupPack	Duplicate-only packets received	Integer32
RcvDupByte	Duplicate-only bytes received	Integer32

RcvPartDupPack	Packets with some duplicate data	Integer32
RcvPartDupByte	Duplicate bytes in part-duplicate packets	Integer32
RcvOutPack	Out-of-order packets received	Integer32
RcvOutByte	Out-of-order bytes received	Integer32
RcvAfterWinPack	Packets with data after window	Integer32
RcvAfterWinByte	Bytes received after window	Integer32
RcvAfterClose	Packets received after close	Integer32
RcvWinProbe	Window probe packets received	Integer32
RcvDupAck	Duplicate ACKs received	Integer32
RcvAckTooMuch	Received ACKs for unsend data	Integer32
RcvAckPack	ACK packets received	Integer32
RcvAckByte	Bytes ACKed by received ACKs	Integer32
RcvWinUpdate	Window update packets received	Integer32
PawsDrop	Segments dropped due to PAWS	Integer32
PredictAcks	Times header predict OK for ACKs	Integer32
PredictData	Times header predict OK for data pkts	Integer32
Pcbcachemiss	Packets missing PCB cache	Integer32
PersistDrop	Timeout in persist state	Integer32
BadSyn	Number of premature ACKs	Integer32
DroppedSyn	Dropped SYNs because of sonewconn() failed	Integer32
ScAdded	Number of SYN cache entries added	Integer32
ScCompleted	Number of SYN cache connections completed	Integer32
ScTimedOut	Number of SYN cache entries timed out	Integer32
ScOverflowed	Number of SYN cache entries dropped due to overflow	Integer32
ScReset	Number of SYN cache entries dropped due to RST	Integer32
ScUnreach	Number of SYN cache entries dropped due to ICMP unreachable	Integer32
ScBucketOvflw	Number of SYN cache entries dropped due to bucket overflow	Integer32
ScAborted	Number of SYN cache aborted (no mem)	Integer32
ScDupeSyn	Number of duplicated SYNs received	Integer32
ScDropped	Number of SYNs dropped (no route/mem)	Integer32

ENTITY TcpQio	QIO	
Error	Collection Error	Integer64
DataMds	Number of data message descriptors (MDs) in use by the process	Integer32
MaxDataMds	Maximum number of data message descriptors (MDs) that have been in use	Integer32
DupMds	Number of duplicate message descriptors (MDs) in use by the process not assigned to inbound driver MDs	Integer32
MaxDupMds	Maximum number of duplicate message descriptors (MDs) that have been in use not assigned to inbound driver MDs	Integer32
NoDataMds	Number of times the process failed to get a data message descriptor (MD)	Integer32
NoDupMds	Number of times the process failed to get a duplicate message descriptor (MD)	Integer32
ResRecovRun	Number of times the send or receive queue on a TCP session exceeded the limit of MDs queued	Integer32
QioLimitWarns	Number of times the process received from the QIO monitor process an event signifying a pool or MD shortage	Integer32
QioDriverErrs	Number of times the QIO driver returned an error	Integer32
CurrPoolSpace	Number of bytes of pool space in use	Integer32
MaxPoolSpace	Maximum pool space used	Integer32
PoolFails	Number of times a pool space request failed	Integer32
TotalMbufs	Number of MBUFs allocated	Integer32
CurrMbufsInuse	Number of MBUFs in use	Integer32
MaxMbufsInuse	Maximum number of MBUFs to be used	Integer32
MbufFails	Number of times an MBUF was not available	Integer32
DupDrivMds	Number of duplicate message descriptors (MDs) allocated	Integer32
MaxDupDrivMds	The maximum number of duplicate message descriptors (MDs) allocated	Integer32
NoDupDrivMds	Number of times a duplicate message descriptor (MD) could not be allocated	Integer32
Size128	Count of socket sends between 1 and 128 bytes	Integer32

Size256	Count of socket sends between 129 and 256 bytes	Integer32
Size512	Count of socket sends between 257 and 512 bytes	Integer32
Size1024	Count of socket sends between 513 and 1024 bytes	Integer32
Size2048	Count of socket sends between 1025 and 2048 bytes	Integer32
Size4096	Count of socket sends between 2049 and 4096 bytes	Integer32
Size8192	Count of socket sends between 4097 and 8192 bytes	Integer32
Size12288	Count of socket sends between 8193 and 12288 bytes	Integer32
Size16384	Count of socket sends between 12289 and 16384 bytes	Integer32
Size32768	Count of socket sends between 16385 and 32768 bytes	Integer32
Size57344	Count of socket sends greater than 32768 bytes	Integer32
QioSolp	Number of SO IP allocations	Integer32
SoForcedClose	Number of SO force closes	Integer32
QioSoQueueFulls	Number of SO queue fulls	Integer32
MdcopyAdj	Number of message descriptor (MD) copy adjustments	Integer32
OutHdrAdj	Number of message descriptor (MD) header adjustments	Integer32
OutdataAdj	Number of message descriptor (MD) data adjustments	Integer32

ENTITY TcpRoute Route

Error	Collection Error	Integer64
RefCnt	Number of users of the route	Integer32
SnetName	Subnet Name	Character 8 wide
ARte	All Route	Character 4 wide
Family	Protocol family	Character 6 wide
RType	Radix type	Character 6 wide
Usage	Number of times the specific route has been used to send out IP datagrams	Integer32

ENTITY TcpRte Process Route Layer

Error	Collection Error	Integer64
BadRedir	Number of bad route redirect requests	Integer32
Dyn	Number dynamic redirects from smart gateways	Integer32
NewGw	Number redirects from new/unknown gateways	Integer32
NotReach	Number of unreachable destinations	Integer32
WildCard	Number of wild card matches found	Integer32

ENTITY TcpSock Socket Migration

Error	Collection Error	Integer64
BcastAllcpusSnt	Broadcasts sent to all CPUs	Integer32
BcastOnecpuSent	Broadcasts sent to one CPU	Integer32
BcastRcvd	Broadcasts received	Integer32
MigrSent	Migration requests sent	Integer32
MigrContSent	Migration continuation requests sent	Integer32
MigrRcvd	Migration continuation received	Integer32
MigrContRcvd	Migration continuation requests received	Integer32
MigrTimedout	Migration timeouts	Integer32
MigrUxpRcvd	Unexpected migrations received	Integer32
MigrContUxpRcvd	Unexpected migration continuations received	Integer32

ENTITY TcpSubnet Subnet

Error	Collection Error	Integer64
Alias	Has Alias	Integer32
Failovr	FAILOVER	Integer32
ShrdIp	Shared IP	Integer32
BroName	Brother name	Integer32
MState	Media State	Integer32
Ipv6	Indicates IPv6 support	Integer32
Ipv4	Indicates IPv4 support	Integer32
OutPkts	Number of packets output on the subnet interface	Integer32
InPkts	Number of packets input on the subnet interface	Integer32
OutErrs	Number of errors detected while outputting packets on the subnet interface	Integer32

InErrs	Number of errors detected while inputting packets on the subnet interface	Integer32
MmacErrs	MMAC Errors	Integer32
MmacTmOts	MMAC Timeouts	Integer32
TcpFlts	TCP filters	Integer32
TcpFltErr	TCP filter errors	Integer32
TcpFltDreg	TCP filters dereg	Integer32
UdpFlts	UDP filters	Integer32
UdpFltErr	UDP filter errors	Integer32
UdpFltDreg	UDP filters dereg	Integer32
PrtFltDrp	Port filters drop	Integer32
DataDrop	Data drops	Integer32
SnetAbort	Subnet abort	Integer32
SnetReset	Subnet reset	Integer32
SnetLstErr	Subnet last error	Integer32
MdaLost	Media lost	Integer32

ENTITY TcpUdp User Datagram Protocol

Error	Collection Error	Integer64
InPkts	Total input packets	Integer32
BadHdrLen	Number of packets received with invalid header size specified	Integer32
BadXsum	Number of packets received with bad checksums	Integer32
NoSum	Number of packets received with no checksum	Integer32
BadLen	Number of packets received with bad packet size	Integer32
NoPort	No socket on port	Integer32
NoPortBcast	No socket on port, arrived as broadcast	Integer32
FullSock	Not delivered, input socket full	Integer32
PcbCacheMiss	Input packets missing PCB cache	Integer32
OutPkts	Number of packets sent to the IP layer	Integer32
InPktsDropped	Number of packets not forwarded to socket applications because of receive socket space being full – HPE NonStop™ specific	Integer32

OutPktsDropped	The number of packets not sent because of interface problems – HPE NonStop™ specific	Integer32
----------------	--	-----------

**ENTITY
TelservProcess**

Error	Collection Error	Integer64
Inconn	The number of incoming Telnet connect requests	Integer32
Open	The number of open requests processed	Integer32
Netdata	The number of bytes received from the network	Integer32
Avgactvterm	The average number of terminals in the system in the last five minutes	Integer32
Spireq	The number of SPI requests processed	Integer32
Userdata	The number of bytes sent by the user to the network	Integer32
Netreq	The number of packets received from the network	Integer32
Userreq	The number of user requests	Integer32

**ENTITY
TelservService**

Error	Collection Error	Integer64
Wins	The number of windows associated with a service	Integer32
Sessions	The number of connection attempts to the service	Integer32
Aborts	Number of connections aborted because client cannot be authenticated, did not have privileges, or no login	Integer32
Inuses	The maximum number of windows in use at sample time	Integer32

**ENTITY
TelservWindow**

Error	Collection Error	Integer64
DataTran	Data transmission bits	Integer32
UserBytes	The number of bytes sent by the user	Integer32
NetBytes	The number of bytes received from the network	Integer32
Reads	The number of read requests posted by the user	Integer32

Writes	The number of write requests posted by the user	Integer32
WriteReads	The number of writeread requests posted by the user	Integer32
Ctls	The number of control requests posted by the user	Integer32
Setmodes	The number of setmode requests posted by the user	Integer32
Cancel	The number of cancel requests posted by the user	Integer32
LAddr	Local IP Address	IpAddr
FAddr	Foreign IP Address	IpAddr
LPort	Local Port	Integer
FPort	Foreign Port	Integer
Servname	First eight characters of the service name	Character 8 wide
WinTyp	Type of Window, Static or Dynamic	Character 7 wide
CAddr	Caller's IP address	IpAddr
CPort	Caller's Port	Integer
LAddr6	Local IP Address	IpAddr6
FAddr6	Foreign IP Address	IpAddr6
CAddr6	Caller's IP address	IpAddr6
Openers	Number of opens against the window	Integer32
Processes	Number of processes found running against the window	Integer32
Active	The number of processes that appear to be logged on and active	Integer32

TMF Entity-Supported Objective Attributes

Attribute	Description	Value
Error	Error number indicating status of data	Integer64
Active	Yes means audit dump is currently dumping	Character 7 wide
Ovflw	Yes means overflow space in use	Character 3 wide
Hold	Yes means no audit because no audit trail space	Character 3 wide
Used	Percentage of audit-trail file used	Integer
Tps	Transaction rate per second	Integer64 (implied decimal)
BeginTr	Begin transaction status	Character 8 wide
Trans	Number of pending transactions	Integer
Hung	Yes means at least one transaction is hung	Integer
Duration	Elapsed time of a transaction in seconds	Integer32

For more information about ASAPX, see the ASAP Extension Manual.

Section 5: File and Process Monitoring

ASAP provides various ways to monitor files and processes and to control the amount of data produced for those objects. The most common method of monitoring a file or process is to specify the Guardian file or process name using the ASAP MONITOR command. For example, the commands `MONITOR FILE $DATA.SUBVOL.FILE` and `MONITOR PROCESS $ABC` will configure ASAP to monitor file `$DATA.SUBVOL.FILE` and process `$ABC`. This section discusses more uncommon ways to monitor files and processes using ASAP, which include:

- Monitoring files and processes using wild-card names
- Monitoring all the processes running from an object file
- Monitoring OSS files
- Monitoring processes executing from an OSS object file
- Grouping files into logical, hierarchical groups of files
- Grouping processes into logical, hierarchical groups of processes
- Defining aggregate and aggregate-only domains at any level of the name hierarchy

Specifying Wildcard Names

You can use the "*" and "?" wildcard characters to specify Guardian file or process names to monitor. Wildcards are not accepted at the volume name level of a filename. Wildcards are only permissible in the subvolume and filename portions.

When wildcard names are specified, ASAP resolves the name at each interval to determine if the files or processes meet the criteria, and ASAP computes and creates a record for each object found. To avoid name conflicts with other forms of name specifiers, ASAP appends the object name to the end of the name specified as a new level of the ASAP domain name. For example, the command `MONITOR PROCESS $A*` would resolve into a domain named "\$A*\$ABC" for process `$ABC`.

Because the set of objects resolved from a wild-card name can vary at each interval, ASAP will not issue an alert if it finds a member of the set is no longer running or present. This is contrary to the way ASAP works if you directly monitor a file or process without using wildcard names. In that case, ASAP does issue an alert when it cannot

find the file or process. ASAP does provide counts of objects when viewing the wildcard domains at aggregate levels.

Wild-card names can resolve into many thousands of objects. That number of records can put pressure on disk resources when storing historical data, so ASAP automatically adds an aggregate-only domain to limit the output from wild-card name specifiers after you issue a MONITOR command that contains a wild-card name. You must manually delete the aggregate-only domain for individual records on each object to be written to the historical database. For example if you enter `MONITOR PROCESS $A*`, then ASAP issues a `MONITOR PROCESS $A*\##` command to add an aggregate-only domain. For more information on aggregate and aggregate-only domains, see the subsections on aggregation later in this section.

Wild-card domains that are resolved at each interval are stored in a memory pool in the File and Process SGPs. The pool is allocated from extended memory and is controlled by KMSF. Excessive numbers of objects can deplete the pool space. The `MONITOR FILE | PROCESS, LIST, DETAIL` command can be used to view pool usage. The `MBYTES` parameter to the ASAP File and Process SGP processes is used to control the amount of space within the pool. For more information, see the [MONITOR Command](#), and the `FILE` and `PROCESS` options of the [SET Command](#).

To set an objective on the wild-card domain, you must include the wild-card specifier in the name. For example, to set an objective on the Full attribute for process \$ABC that was resolved from the \$A* wild-card specifier, you enter `RANK PROCESS $A*\$ABC, FULL < 90`. You can also set a global objective at the wildcard level, for example `RANK PROCESS $A*, BUSY < 95`.

Monitoring Processes from an Object File

ASAP supports monitoring of all processes running from an object file. The object file can be a Guardian or OSS object filename. When object filenames are specified, ASAP resolves the name at each interval to determine the processes that are executing from that object file. ASAP computes and creates a record for each process found. To avoid name conflicts with other forms of name specifiers ASAP appends the process name to the end of the name specified as a new level of the ASAP domain name. For example, the command `MONITOR PROCESS $DATA.OBJECT.FILE` would resolve into a domain named `$DATA.OBJECT.FILE\$ABC` for process \$ABC.

Because the set of objects resolved from an object filename can vary at each interval, ASAP does not issue an alert if it finds a member of the set is no longer running or present. This contrasts the way ASAP works if you directly monitor a process using the process name. In that case ASAP does issue an alert when it cannot find the process. ASAP does provide counts of objects when viewing the domains at aggregate levels.

Object filenames can resolve into many objects. That number of records can put pressure on disk resources when storing historical data, so ASAP automatically adds an aggregate-only domain to limit the output from object-filename specifiers after you

issue a MONITOR command that contains an object filename. A user must manually delete the aggregate-only domain for individual records on each object to be written to the historical database. For example if you enter `MONITOR PROCESS $DATA.OBJECT.FILE`, then ASAP issues a `MONITOR PROCESS $DATA.OBJECT.FILE\##` command to add an aggregate-only domain. For more information on aggregate and aggregate-only domains, see the sections on aggregation later in this section.

Object filename domains that are resolved at each interval are stored in a memory pool in the Process SGP. The pool is allocated from extended memory and is controlled by KMSF. Excessive numbers of objects can deplete the pool space. The `MONITOR PROCESS, LIST, DETAIL` command can be used to view pool usage and the `MBYTES` parameter to the ASAP Process SGP process is used to control the amount of space within the pool. For more information, see the [MONITOR Command](#) and `PROCESS` option in the [SET Command](#).

To set an objective on a domain resolved from an object filename you must include the object filename specifier in the name. For example, to set an objective on the Full attribute for process \$ABC that was resolved from the \$DATA.OBJECT.FILE object filename specifier, enter `RANK PROCESS $DATA.OBJECT.FILE\ $ABC, FULL < 90`. You can also set a global objective at the object filename level; for example, `RANK PROCESS $DATA.OBJECT.FILE, BUSY < 95`.

Note

Resolving the processes running from an object filename can result in excessive low-level messaging in the NonStop™ OS. This method should be used judiciously in extremely high-performance application environments.

Monitoring Files and Processes Using OSS Pathnames

OSS pathnames can be specified as domain names or separately using the `PATH` option to the `MONITOR` command when monitoring OSS files and processes.

Because OSS pathnames can be longer and can contain more levels than ASAP domain names, ASAP truncates any OSS pathname specified as a domain name when the entire domain name exceeds 64 bytes or contains more than 5 levels. ASAP truncates the leftmost portion of the pathname and includes a partial filename but not the partial directory levels. Only entire directory levels are included, and only when the level does not exceed the maximum level requirement.

ASAP transforms the pathname portion of a domain name into proper ASAP domain name form, using the backslash as the level separator instead of a slash. When you enter `MONITOR PROCESS /bin/sh` and ASAP finds a process running from the `/bin/sh` object file, it creates a new record for that process by adding the process identifier as a new level. An actual domain that would be displayed in an ASAP record would be

Bin\Sh\\$\$Name for a process named \$Name or might be Bin\Sh\\$.2.59.6181274 for the unnamed process running in CPU 2, PIN 59.

The OSS pathname can be the complete domain name, as shown in the /bin/sh example, or it can be preceded by a logical name as described in the next section. For example, MONITOR PROCESS OSS\SHELL/bin/sh creates an ASAP domain name of OSS\SHELL\BIN\SH. The name OSS\SHELL/bin/sh is equivalent to OSS\SHELL\bin/sh.

If OSS pathnames cannot work as a part of an ASAP domain name, or you do not like the way ASAP converts the pathname into a domain name, you can create your own, unique domain name and use the MONITOR command PATH option to specify the OSS pathname.

All OSS pathnames must begin with a slash.

Note

Resolving the processes running from an OSS pathname can result in excessive low-level messaging in the NonStop™ OS. This method should be used judiciously in extremely high-performance application environments.

Logical Grouping

ASAP lets you group files into hierarchical groups of files and processes into hierarchical groups of processes. This feature is implemented in the MONITOR command, allowing you to specify a logical group name or names that the file or process belongs to. For example, these monitor commands define processes \$A10 through \$A14 as members of the Sales application:

```
Monitor Process Sales\Order\Entry\$$A10
```

```
Monitor Process Sales\Order\Entry\$$A11
```

```
Monitor Process Sales\Order\Fullfillment\$$A12
```

```
Monitor Process Sales\Mfg\Build\Westcoast\$$A13
```

```
Monitor Process Sales\Shipments\International\$$A14
```

The rules for the logical domain names are:

1. The entire name must be between 2 through 64 bytes in length.
2. The name can contain up to five levels separated by a backslash.
3. The last level of the name must be an object selector: a filename or process name, a wildcard file or process name, or a process object filename.
4. The name can be constructed with any printable ASCII character except a space, colon, semicolon, quotation mark, or asterisk.
5. A physical file or process can belong to multiple groups.

Note

In this case, each file or process instance is monitored separately by ASAP and could produce slightly different results depending on timing and other factors.

Domain Aggregation

ASAP lets you aggregate File and Process domains at any level of the hierarchical group name. ASAP can automatically construct aggregate domains for all objects at all levels using a global parameter setting, or you can define only the necessary aggregate domains for individual groups (recommended). Aggregate domain names are constructed using the # character as the level name. For example, the aggregate domain Sales\# represents the combination of all domains in the Sales group. Similarly, Sales\Orders\# represents all members of the Sales\Order group.

There are two ways to turn on aggregation for files and processes being monitored as a part of a hierarchical group.

Creating Specific Aggregate Domains (Recommended)

To create an aggregate domain for hierarchically grouped files or processes, use the MONITOR command to add the aggregate domain to the ASAP configuration. For example, MONITOR PROCESS SALES\# causes ASAP to create the Sales\# domain and to aggregate all domains starting with Sales into that domain.

Setting Global Aggregates for All Domains

An ASAPCONF parameter setting for File and Process, AGG, causes ASAP to automatically create aggregate records at each level of any hierarchical file or process name. If AGG is defined, you need not add aggregate domains to the database. ASAP creates them automatically.

Aggregate Only Records

When monitoring thousands of files or processes, limit the amount of historical data being stored by ASAP by configuring ASAP to write only aggregate domain records to the database, leaving detail records in memory for command-based retrieval when necessary. This solution can result in significant data reduction while still accurately recording usage information for historical purposes.

When an aggregate domain issues an alert because a member domain has alerted, ASAP provides the ability to retrieve information about the alerting file or process directly from ASAP memory. For more information, see the MEMORY option for the [FILE Command](#) and the [PROCESS Command](#).

There are two ways to turn on aggregation for objects being monitored as a part of a hierarchical group and to write only the aggregate records to the ASAP database.

Creating Specific Aggregate-Only Domains (Recommended)

To cause ASAP to write only the aggregate record for a set of domains, use the MONITOR command to add an aggregate domain using ## as the last level specifier. For example, the command MONITOR PROCESS SALES\## causes ASAP to write only the Sales\## record to disk. ASAP still computes data for each individual process and then combines the data into the aggregate record, but it writes only the aggregate record to disk.

Setting Global Aggregate Only for All Domains

An ASAPCONF parameter setting for File and Process, AGGONLY, causes ASAP to automatically create aggregate records at each level of any hierarchical process name and to write only the aggregate domains to the database. If AGGONLY is defined, you need not add aggregate domains to the database. ASAP automatically creates them.

Aggregation Defined

The way attribute values are combined in the aggregate records varies by attribute and can be changed. Different users might want different behavior for a specific attribute. For example, consider the Process Busy, Pri and PFName attributes. (Busy represents process busy.) By default, ASAP sums all the process busy values in the group for the aggregate record, so you can see the total consumption of processor resources across the group.

However, it does not make sense to sum the values of the Pri attribute, which represents the priority of the processes. ASAP assumes you are interested in the process with the lowest priority, by default, so it uses the minimum Pri attribute value found among all processes in the group when creating the Pri value in the summary record.

Also consider the PFName attribute that contains the object file name for each process. It does not make sense to perform any calculations on that attribute in the summary record. For PFName and several other process attributes, ASAP propagates the value to the summary record from the process with the highest (worst) ASAP state. For example, if there are 100 processes, with two issuing a critical alert (ASAP state 7) and one issuing a down alert (state 8), ASAP propagates the PFName value from the down process to the summary record. If multiple processes have the same worst state, the first one found is propagated to the summary record.

ASAP gives you control over how data is ultimately combined in the aggregate record. See the next subsections.

State Propagation

ASAP always propagates the worst state for each attribute to the aggregate record regardless of how the attribute value is computed in the aggregate record unless

specific objectives are set on the attribute at the aggregate level. The state associated with the value in an aggregate record might or might not reflect the actual state of the value shown. Consider the Process Busy attribute. In the aggregate record, ASAP combines all the process busy attribute values in the group to show the total processor consumption of the group. However, the state shown for that attribute, unless overridden by setting a specific objective at the aggregate level, represents the worst state found for the busy attribute among all the members.

For example, consider a group of five processes, Sales\A10 through Sales\A15. ASAP combines all five process busy attributes when it creates the Sales\# domain, showing the total consumption of all five processes. However, if process A13 is issuing an alert because it is too busy, that alert level is propagated to the aggregate domain. Setting specific objectives on an aggregate domain overrides state propagation. For example, `RANK SALES\#, BUSY < 20` sets an objective on the Busy attribute at the aggregate level, and the result of that objective comparison produces the ASAP state for Busy, not the propagated state value.

Aggregation Rules

The EDL Attribute property MetricRule is used to define the aggregation property of a File or Process attribute. You can use these values:

SUM	Sum the value across all members of the aggregate grouping
AVG	Average the value across all members of the aggregate grouping
MIN	Use the minimum value found in the aggregate grouping
MAX	Use the maximum value found in the aggregate grouping
CNT	Use a count of the number of values found in the aggregate group
RUL	Use the ASAP rule for PState and WState. See Process PState and WState Attributes

The EDL definition for the Process Busy attribute is:

```
AT Busy Grid YES Graph YES GraphMax 0
Format "F6.2"
Help "Process Busy"
MetricRule SUM
StatePair YES
StateRule UseStateGraphState
TypeData REAL64;
```

The statement MetricRule SUM defines the aggregation rule, to sum all members of the group. This statement can be changed to use AVG, MIN, MAX or CNT instead of the summary, or the statement can be removed to cause the value associated with the worst state to be propagated to the aggregate record.

Attribute Propagation

If no aggregation rule is defined for an attribute, the value associated with the attribute with the worst state is also propagated. For example, if you remove the SUM rule from the Process Busy attribute in the previous example, the Busy value for process Sales\A13 is propagated to the aggregate record instead of the sum of all processes.

By default, all File attribute values are propagated to the aggregate record. By default, Process attribute values vary based on the MetricRule property.

Status and OpState

The Status attribute is handled differently than other attributes in an aggregate record. The Status text field is used to show the number of members of the group and the number that are issuing alerts. For example, if there are five files in the group and there are no alerts, the Status text reads "5 Files." If there are five processes in the group with one issuing alerts, the text reads "5(1) Processes."

The state at which ASAP considers a file or process to be alerting is the Low Alert state 3, so if any attribute in the domain record has a state of 3 or higher, the domain is counted in the number of alerting domains shown in the Status text field. If you prefer a different alert level, you can change the state at which ASAP considers a process to be alerting. See the FILE and PROCESS options for the [SET Command](#).

ASAP Alert Levels:

- | | |
|---|--------------|
| 1 | Exists |
| 2 | OK |
| 3 | Low Alert |
| 4 | Medium Alert |
| 5 | High Alert |
| 6 | Warning |
| 7 | Critical |
| 8 | Down |

The associated state for Status, OpState, is propagated from the worst state within the set like all other state attributes.

Attributes without States

Many attributes, such as the Process PFName attribute, have no associated state. You cannot define an objective value on those attributes, and ASAP does not rank them. For these attributes, ASAP propagates the values from the domain with the highest alert level.

ASAP propagates these attributes regardless of any MetricRule setting for the attribute:

File Attributes

Code Format Type Partitions LastOpen LastModif

Process Attributes

BPin Caid CLoc CreatedPaid PFName

PFS Pid Pin SMod SReq Type

Process PState and WState Attributes

The process state and wait state attributes cannot be combined using aggregate math because each represents the unique state of an individual process. Typically, these values are propagated to the aggregate record using the value from the attribute with the worst state, but they can also be aggregated based on specific priority rules. This behavior can be turned on with the RUL aggregate control and results in this priority ranking of the possible values, from low to high:

PState	WState
UnAlloc	None
Starting	Lreq
Runnable	Lpipe
DebugBrk	Pon
DebugTrp	lopon
DebugReq	Intr
InspMBrk	Lcan
InspBrk	Ldone
InspTrp	Ltmf
InspReq	Linsp
SaveAben	

Stopping
XIOInit

As an example, when RUL is specified, a process in the Stopping process state is considered to be worse than a process in the Runnable process state, so the Stopping value is propagated to the aggregate record based on the priority rule. Like other attributes, the worst associated state value is always propagated to the aggregate level.

Default Process MetricRule Settings

This table shows the MetricRule aggregate controls that are set for Process monitoring when you first install ASAP. Any other Process attributes have no default setting and are state propagated:

Attribute	Default Value
Pri	MIN
Busy	SUM
PState	RUL
WState	RUL
Mpages	SUM
Msent	SUM
MRecv	SUM
QLen	MAX
MQLen	MAX
Pfaults	SUM

The Process Count attribute provides an automatic count of all domains in the group and is always one for a detail domain.

The Process ACount attribute provides the number of attributes that are alerting when viewed in a detail process record and provides the number of domains that are alerting when viewed in an aggregate domain record.

Modifying the Default Aggregation Settings

To change the default Process aggregation controls provided in ASAP, edit the ASAP2SYS EDL file, and copy the File or Process EDL definition into a separate file. Make changes to the new file and then add an INCLUDE statement to the ASAPUSER file to include the new File or Process EDL file. Change only the MetricRule formulas for attributes. Any other EDL changes can produce unpredictable results.

Section 6: ASAP Conversational Interface Commands

The ASAP Conversational Interface (CI) provides the commands and options to start, stop, configure, and get the status of the ASAP system, as well as the commands to display object status and performance data.

Use the CI to configure, start, stop, and get the status of the system resources Application (APP), CPU, DISK (DSK), Expand (LH), FILE(FIL), Node end-to-end (NCP), PROCESS(PRO), executing ProcessBusy (PCB), RDF(RDF), SPOOLER (SPL), TAPE(TAP), and TMF(TMf).

The CI displays performance reports for Application, CPU, Disk, Expand, Node, and Process objects. Default performance reports contain object status information about Application, CPU, Disk, Expand, File, Process, RDF, Spooler, Tape, and TMF.

To start the ASAP CI, type `ASAP` at a TACL prompt. The startup syntax is:

```
ASAP / run-option [, run-option ] .. / command
```

run-option

specifies one or more of the standard NonStop™ OS startup options. For more information, see the TACL Reference Manual.

IN *filename*

specifies the command input file.

OUT *filename*

specifies the output file.

command

specifies one of the commands in this section. If the ASAPCONF file is on the current default subvolume or in the \$SYSTEM.SYSTEM subvolume, ASAP automatically obeys it before executing a command.

This section discusses the ASAP commands and provides detailed information about each command including:

- The purpose and function of the command
- The complete syntax for the command

- A description of all available command options
- Examples illustrating the use of each command

Table 6-1. ASAP Command Summary

Command	Function	Page
ACTION	Defines action rules and lists defined actions	144
APP	Displays performance and state information for ASAP-enabled applications	147
CLEANUP	Manually invokes the ASAP database cleanup	155
COMM	Displays performance and state for all monitored SWAN communication lines	158
COMMIT	Instructs ASAP to commit all database changes	163
CPU	Displays performance and state information for each CPU	163
DB	Displays information about ASAP data base file disk space utilization when the SET DBMAXDAYS days option is enabled.	
DISK	Displays performance and state information for all nonphantom disks	171
EDL	Performs EDL syntax check (SET EDL compiles EDL)	178
EXIT	Stops execution of the ASAP CI	179
FC	Invokes the standard FIX command	179
FILE	Reports on file statistics	179
GOAL	Controls the objectives ASAP uses to produce alerts	184
HELP	Displays an explanation of the specified ASAP command	198
LH	Displays Expand line-handler performance and state information for all Expand lines	199
LOG	Displays ASAP event log records	208
MONITOR	Controls the set of objects monitored by ASAP	208
NODE	Displays process information for end-to-end Expand nodes	221

OBEY	Execute the commands contained in the specified OBEY file	224
PAUSE	Suspends the program until a stop, abend, or break message is received	224
PROCESS	Displays statistics for the selected processes.	225
PROCESSBUSY [PB]	Displays process performance information for the busiest processes in the specified CPUs and nodes	228
RANK	Controls the objectives ASAP uses to produce alerts	233
RDF	Reports on Remote Database Facility statistics	245
SET	Displays or alters data collection and monitoring options	251
SHOW	Displays entity information	292
SHUTDOWN	Stops the Monitor, SGPs, and Collect servers on specified nodes	293
SPOOLER	Reports on Spooler statistics	294
SWAP	Reports on KMSF statistics	279
STARTUP	Starts the Monitor, SGPs, and Collect servers on specified nodes	299
STATUS	Provides status information about the Monitor, SGPs, and Collect processes	301
SYNC	Manually requests Monitor to synchronize time in the specified nodes	306
SYSTEM	Specifies a default system name	311
TAPE	Reports on Tape statistics and tape mount requests	312
TCP	Reports on TCP/IP and Telserv statistics.	315
TMF	Reports on TMF statistics	318
VOLUME	Displays or changes the default volume name	322
!	Is the comment character. All text prior to the ! is interpreted as a command or option.	

Considerations

- Online help is available for all ASAP commands and options. In ASAP, enter `HELP command`.
- To delimit an end-of-line comment, use the exclamation point (!) before the comment.
- To execute a command, enter enough characters to uniquely identify that command. For example, to execute the STATUS command, enter `STAT. STA`

alone cannot be distinguished from the STARTUP command.

- To continue a command on a subsequent line, end the current line with the ampersand (&) character and enter the remaining portion of the command on the following line. The ASAP CI prompt changes to +& to indicate that it is operating in continuation mode. For example:

```
+SET DISK OBJECT &  
+&$SYSTEM.SYSTEM.ASAPDSK, PARAMETERS "RATE 10"
```

ACTION Command

The ACTION command is used to list action strings for a specific domain or domains and to define, delete and list rules, or stored action strings that can be used in GOAL and RANK commands to define actions for failed objectives. Please see the GOAL or RANK commands for more information on actions.

```
ACTION[/OUT file/] [LIST] [\node][entity][domain][attribute [operator]]  
[LIST] [\node][entity][domain][gname] [, GOALNAME]  
RULE [\node][rule][=string] |[, DELETE]
```

LIST

The LIST option is used to output information about all actions defined in the objectives database. When the entity, domain and attribute parameters are all supplied, the output is returned in tokenized form. LIST is the default when no option is supplied.

\node

The NonStop™ OS node name the action is defined for.

entity

An entity name; for example, CPU, DISK, SPOOLER, TAPE.

domain

A domain name; for example, \$DATA or APP\SALES\DOMAIN.

attribute

An attribute defined for the entity; for example, Busy, Full. Using *attribute* without specifying *operator* can also result in output for actions associated with named Boolean goals if goal names are the same as attribute names.

operator

A relational operator: GT, LT, EQ, NE, GE, or LE. The operator is not specified for the STATUS attribute.

gname

The name of a Boolean goal. Outputs the action defined for this goal name. Using *gname* without specifying GOALNAME can also output attribute specific actions if the action name is the same as an attribute name.

GOALNAME

Specifies that output will be produced in tokenized form for a specific Boolean goal named *gname*.

RULE

The RULE option is used to list, define, and delete stored action strings that can be referenced in RANK commands.

rule

A stored action string name. The name is optional when listing rules but is required to define or delete a rule.

string

An action string not enclosed in quotes, where the first word in the string represents the command interpreter the command should be sent to and the remainder of the string is the command to be sent to the CI.

The string should be entered exactly as it will be sent to the command interpreter. Include any of the tokens below in the string, and ASAP will substitute actual names for the symbolic tokens. For example, this action string can be used to resecure any file that is not meeting its security objective:

```
FUP SECURE <#OBJECT>, "<#GOAL>"
```

Tokens include:

<#ENTITY>	The entity name
<#OBJECT>	The NonStop™ OS or OSS object name
<#DOMAIN>	The domain name
<#D1>	Leftmost part of a multilevel domain name
<#D2>	2nd part of a multilevel domain name
<#D3>	3rd part of a multilevel domain name
<#D4>	4th part of a multilevel domain name
<#D5>	5th part of a multilevel domain name
<#ATTR>	The attribute name
<#OP>	The objective operator

<#VALUE>	The current value of the attribute
<#STATE>	The current state of the attribute
<#LASTSTATE>	The last state of the attribute
<#GOAL>	The objective value
<#GNAME>	The name of the Boolean goal
<#GOALVAL>	Attribute, operator, goal and value in a string with no spaces where goal and value are separated by the character
<#attrGOAL>	A specific goal token for the attribute
<#attrOP>	A specific goal token for the attribute's operator
<#attrVALUE>	A specific goal token for the attribute's value

DELETE

Deletes the referenced action rule.

Examples

ACTION shows all action strings.

ACTION CPU shows all CPU action strings.

ACTION \REMOTE shows all actions on \REMOTE.

ACTION RULE lists all actions stored as rules.

ACTION RULE RESECURE lists the stored action named RESECURE.

To define a stored action to resecure files:

```
ACTION RULE SECURE = FUP SECURE <#OBJECT>, "<#GOAL>"
```

```
ACTION RULE RESECURE, DELETE deletes the RESECURE action.
```

```
ACTION CPU 0 BUSY GT gets action/wait for CPU 0 BUSY GT.
```

```
ACTION CPU 1 CPUGOAL, GOALNAME gets action/wait for CPUGOAL.
```

Default Rules

ASAP automatically loads several rules into its database of Action Rules when it creates it for the first time. These may be used in any GOAL or RANK command.

ACTION RULE ACTIVATE	=	TACL ACTIVATE <#OBJECT>
ACTION RULE ALTPRI	=	TACL ALTPRI <#OBJECT>, <#GOAL>
ACTION RULE DEFRAG	=	DCOM <#OBJECT>
ACTION RULE GIVE	=	FUP GIVE <#OBJECT>, <#GOAL>
ACTION RULE PRIMARY	=	SCF PRIMARY <#OBJECT>, <#GOAL>

ACTION RULE RELOAD	=	TACL RELOAD <#OBJECT>, PRIME
ACTION RULE SECURE	=	FUP SECURE <#OBJECT>, <#GOAL>
ACTION RULE SUSPEND	=	TACL SUSPEND <#OBJECT>

Considerations

Actions requiring multiple interactions with command interpreters or decision logic should be written as TACL macros or be sent to custom programs. Check that TACL macros are loaded and available for ASAP by loading them in the default TACLCTM file for the ASAP user id.

APP Command

The APP command displays customer application service-level metrics for the specified nodes. For customer-developed applications to participate in the ASAP system, ASAPX must be installed. Customer-developed applications must be ASAP enabled through calls to the ASAPX library. For more information, see the ASAP Extension Manual . For an explanation of APP performance statistics, enter `HELP APPSTATS`. To alter the APP output, enter `HELP FORMAT` or see the [HELP Command](#).

```

APP  [/OUT file/] [ \* ] [ [.] * ] [ , SAMPLES count ]
      [ \node ]      [ , CPU ]
                        [ , STATES ]
                        [ , TIME hh[:mm [m/d/y]] ]
                        [ , AGGREGATE ]
                        [ , AGGREGATEONLY ]
                        [ , MINSTATE [AUTO ] ]
                          [ state ] ]
                        [ , COUNT count ]
                        [ , MEMORY ]
                        [ , AGGONLY ]
                        [ , EXTRACT filename ]
                        [ , NOPRINT ]
                        [ , NOLINEBREAK ]
                        [ , NOAGG ] ]

      [ [.] domain ] [ , DETAIL ]
                        [ , SAMPLES count ]
                        [ , CPU ]
                        [ , STATES ]
                        [ , TIME hh[:mm [m/d/y]] ]
                        [ , AGGREGATE ]
                        [ , AGGREGATEONLY ] ]

```

[, MINSTATE [AUTO]]
	[state]]
[, COUNT <i>count</i>]
[, MEMORY]
[, AGGONLY]
[, EXTRACT <i>filename</i>]
[, NOPRINT]
[, NOLINEBREAK]
[, NOAGG]

OUT file

specifies the spooler or line printer where the ASAP output is to be sent. When no output file is specified, the output is displayed on your terminal or workstation.

specifies that the performance and state information for all systems is to be displayed.

node

specifies the individual system for which statistical information is to be displayed. If you want information for all the systems, specify APP ***. If you do not specify a node or the wild-card character (***), the default is used. The default is the system where the ASAP program was started, or the system specified in the ASAP SYSTEM command.

domain

specifies the individual domain name for which customer application performance and state information is displayed. A domain name is a unique logical name that can be composed of up to five component levels, each component level separated by the *-||* character. All of the domains from all of the collected nodes can be output by specifying APP **.**. If you do not specify domain name or the wild-card character (***), the default is used. The default is all of the Application domains in the current system.

The entity name as defined in EDL must be the first part of a domain name, and must be supplied to show DETAIL statistics for any domain. For example, APP **ATM, DETAIL* shows detailed statistics for all ATM domains on all nodes.

Note

If you do not specify an entity name in an APP command, the default APP entity is used to display information. Since APP has no custom attributes defined, you do not see detailed data for any domains. For example, APP *AT* displays ATM domains (and any other domains beginning with AT) but does not show detailed information.

*

specifies that the performance and state information for all domains on the specified system are to be displayed. If you specify a domain or the wild-card character (*), statistics for all of the domains in the current system are displayed.

SAMPLES *count*

specifies the number of sample periods to be displayed. A sample period is defined by the RATE option in the SET command. You can specify 0 through 255 sample periods. The default sample period number is 1.

DETAIL

displays user-defined attributes for the specified entity.

CPU *cpu-number*

specifies that the performance information for the APP domains executing in the specified CPU, *cpu-number*, are displayed. If you do not specify a CPU, statistics for all APPs in the current system are displayed.

STATES

displays the associated OEM state for each attribute where StatePair is true. These OEM states are reported, depending on the type of ranking used (see the SET OBJECTIVESRANK option of the [SET Command](#)):

Existence	1	Object exists.
Up or OK	2	Object is up or meets objectives.
Low	3	Object utilization is low.
Medium	4	Object utilization is medium.
High	5	Object utilization is high.
Warning	6	Object is at Warning level.
Critical	7	Object is at Critical level.
Down	8	Object is down or not operational.

TIME *hh:mm m/d/y*

defines the ending time of the display information in hours, minutes, month, day, and year. For example, specifying TIME 10:15 when RATE is 5 indicates that statistics collected from 10:10 to 10:15 are displayed.

hh

is the hour.

mm

is the minute.

m/d/y

is the month, day, and year. Valid *m/d/y* entries are *mm/dd/yy*, *yyyy/mm/dd*, or *mm/dd/yyyy*.

AGGREGATE

specifies that aggregate records be output along with the detail records. AGGREGATE is on by default.

AGGREGATEONLY

specifies that only aggregate records are output in the command. AGGREGATEONLY is useful in large domain populations for controlling the amount of data sent to the ASAP Client.

MINSTATE [AUTO|*state*]

specifies that only records which have at least one attribute with a state greater than or equal to *state* are to be output. *state* must be a valid OEM state from 1 through 8. AUTO or no *state* specifies to output all records with at least one attribute with a state greater than 2. MINSTATE is useful in EDL DETAIL commands to control output to the ASAP Client.

COUNT *n*

specifies that a maximum of *n* records will be output. COUNT is useful in EDL DETAIL commands to control output to the ASAP Client.

AGGONLY

is an abbreviation for the AGGREGATEONLY option. It can also be abbreviated as AO.

MEMORY

fetches APP data from memory instead of the database. The MEMORY option, in conjunction with MINSTATE and COUNT, should be used in the EDL DETAIL command for entities where SET AGGREGATE ONLY has been defined in ASAPXCNF.

Note

MEMORY does not work with SAMPLES, TIME, or *.

EXTRACT

reads application records from the ASAP database and writes them to another file or process.

NOPRINT

suppresses terminal output during EXTRACT operations.

NOLINEBREAK

instructs ASAP to stop breaking output lines at 132 characters. Use this option when creating CSV file output from ASAP CI and directing the output to a file type or process than can accept long lines without line breaks.

NOAGG

suppresses displaying of aggregate domains.

Examples

These examples show some available options along with the display output:

```
+APP \CENTDIV ATM                                !DEFAULT OUTPUT Domain = ATM
\CENTDIV Domain\Name\Hierarchy  Date  Time  Ver CpuPin  Ct Error Nak UnAvail
-----
Accounts\Payable\Tt29          2/25 12:43 A1 1,121 4 0 0 0.00
Accounts\Payable\Tt30          2/25 12:43 A1 0,130 4 0 0 0.00
Accounts\Receive\Tt31          2/25 12:43 A1 1,122 4 0 0 0.00
Accounts\Receive\Tt32          2/25 12:43 A1 0,131 4 0 0 0.00
```

```
\CENTDIV Domain\Name\Hierarchy  Status          Date  Time  Ct Error
-----
Atm\Guest\Chicago\Jx00          Up              5/10 16:56 17 0
Atm\Guest\Chicago\Jx01          Up              5/10 16:56 17 0
Atm\Guest\Chicago\Jx20          Up              5/10 16:56 17 0
Atm\Guest\Chicago\Jx21          Up              5/10 16:56 17 0
```

```
+APP \CENTDIV MARKET
\CENTDIV Domain\Name\Hierarchy  Status          Date  Time  Ct Error
-----
Market\Commodities\Corn\Frozen  Up              5/10 16:58 12 0
Market\Commodities\Corn\Futures Up              5/10 16:58 12 0
Market\Commodities\Porkbellies\Futures
Up              5/10 16:58 12 0
Market\Commodities\Porkbellies\Salted
Up              5/10 16:58 12 0
Market\Commodities\Soybeans\Futures
Up              5/10 16:58 12 0
\CENTDIV Domain\Name\Hierarchy  Date  Time  Ver CpuPin  Ct Error Nak
UnAvail
-----
```



```
Test\Onereallongname\Domaintypename\#Nextlongsubdevice.name
                2/25  12:43  A1  0,115   4   0   0   0.00
```

The preceding example shows basic APP command output without detail attributes, and it also shows how ASAP wraps to a second line when the domain name is longer than 33 characters. The domain name appears on its own line, and its corresponding data appears on the next line.

The following example, of the Accounts application, uses the DETAIL option so the output displays the four user-defined metrics: *TransRate*, *ErrCount*, *Busy*, and *State*.

The EDL definition for the Accounts entity shows the user-defined attributes:

```
ENTITY Accounts  CI          ASAP,
    COMMAND      "APP\*ACCOUNTS,RAW,TAB,DETAIL,STATE,AGGREGATE" ,
    DETAIL       "APP\*ACCOUNTS,TAB,DE,ST,AG,MI,COUNT 25"
    DataItems    "0 I, 1 I, 2 U, 3 M, 4 S, 5 C"
    ENABLED      YES,
    ErrorState   ErrorState
    KeyForNode   NodeName, KeyForObj Domain, KeyForRow "Dateymd Time",
    RESERVED    NO
    SGPManage    YES, SGPFile "$System.System.ASAPXSGP", SGPSuffix H
    HELP        "TRANSACTION Application", MaxObjectives 100;

AT NodeName    GRID YES    GRAPH NO    GRAPHMAX 0    HELP "NSK System Name";
AT Sysnum      GRID NO     GRAPH NO    GRAPHMAX 0    HELP "System Number";
AT Domain      GRID YES    GRAPH NO    GRAPHMAX 0    HELP "Domain Name";
AT Status      GRID YES    GRAPH YES    GRAPHMAX 0    HELP "Operational Status"
    HEAD OpStatus HeadPickup True
    STATEPAIR YES STATERULE UseStateGraphState STATEISOP YES
    TypeData CHAR20;
AT OpState     GRID NO     GRAPH NO    GRAPHMAX 9    HELP "Objects Operational
State";
AT Dateymd     GRID NO     GRAPH NO    GRAPHMAX 0    HELP "Date of Stats";
AT Time        GRID YES    GRAPH YES    GRAPHMAX 0    HELP "Time of Stats";
AT Valid       GRID NO     GRAPH NO    GRAPHMAX 0    HELP "Validity Flag";
AT ET          GRID NO     GRAPH NO    GRAPHMAX 0    HELP "Elapsed Time in Minutes";
AT CT          GRID NO     GRAPH NO    GRAPHMAX 0    HELP "Count of Attributes";
AT Error       GRID YES    GRAPH NO    GRAPHMAX 1    HELP "Collection Error"
    STATEPAIR YES STATERULE UseStateGraphState
    TypeData INT64
    Format "I4";
AT ErrorState  GRID NO     GRAPH NO    GRAPHMAX 9    HELP "State of Error";
```

```

AT TransRateGRID YES          GRAPH YES  GRAPHMAX 10 HELP "Transaction rate"
    STATEPAIR YES STATERULE UseStateGraphState
    TypeData REAL64
    MetricRule "#0/S"
    Format "F8.4";
AT S0 GRID NO    GRAPH NO    GRAPHMAX 9   HELP "State of TransRate"
    HEADING "S";

AT ErrCount      GRID YES  GRAPH YES  GRAPHMAX 10 HELP "Error count"
    STATEPAIR YES STATERULE UseStateGraphState
    TypeData REAL64
    MetricRule "#1"
    Format "I3";
AT S1  GRID NO    GRAPH NO    GRAPHMAX 9   HELP "State of ErrCount"
    HEADING "S";

AT Busy          GRID YES  GRAPH YES  GRAPHMAX 10 HELP "Percent Busy"
    STATEPAIR YES STATERULE UseStateGraphState
    TypeData REAL64
    Format "F5.2";
AT BusyStateGRID NO    GRAPH NO    GRAPHMAX 9   HELP "State of Busy"
    HEADING "S";

AT State        GRID YES  GRAPH YES  GRAPHMAX 10 HELP "Server state"
    STATEPAIR YES STATERULE UseStateGraphState
    TypeData REAL64
    MetricRule "#5"
    Format "I1";
AT S3 GRID NO    GRAPH NO    GRAPHMAX 9   HELP "State of State"
    HEADING "S";

```

```

+APP \CENTDIV ACCOUNTS,DETAIL
\CENTDIV Domain\Name\Hierarchy   Status           Date   Time   Ct  Error
-----
Payable\$Jx48                     Up                5/10 17:41   4    0
Payable\$Jx49                     Up                5/10 17:41   4    0
Receive\$Jx50                     Up                5/10 17:41   4    0
Receive\$Jx51                     Up                5/10 17:41   4    0

```

continued...

```

TransRate ErrCount Busy  State
-----
0.3000      2  0.05   5
0.7500      4  0.12   1
0.3000      3  0.07   4
1.2000      8  0.20   4

```

```

\CENTDIV Domain\Name\Hierarchy
Date   Time  Ver  CpuPin Ct  Error Nak  UnAvail TransRate ErrCount Busy  State
-----
Accounts\Payable\$Tt29
2/25 12:43 A1  1,121  4  0    0  0.00    1.33    16  0.12  3
Accounts\Payable\$Tt30
2/25 12:43 A1  0,130  4  0    0  0.00    0.67     8  0.06  7
Accounts\Receive\$Tt31
2/25 12:43 A1  1,122  4  0    0  0.00    0.50     6  0.03  1
Accounts\Receive\$Tt32

```

```
2/25 12:43 A1 0,131 4 0 0 0.00 1.33 16 0.13 5
```

```
+APP \CENTDIV ACCOUNTS\RECEIVE,DETAIL
```

\CENTDIV Domain\Name\Hierarchy	Status	Date	Time	Ct	Error
Receive\Jx50	Up	5/10	17:47	4	0
Receive\Jx51	Up	5/10	17:47	4	0

```
continued...
```

TransRate	ErrCount	Busy	State
0.1500	2	0.04	3
0.6000	4	0.10	3

This example retrieves detail record information about the ATM application directly from ASAP memory instead of from the ASAP database.

```
APP ATM, DETAIL, STATE, MEMORY
```

This example limits the response to an in-memory record request only to domains where at least one attribute state is at the medium alert level (state 4.)

```
APP ATM, DE, ST, MEM, MINSTATE 4
```

This example shows information from the ASAP database but limits the output to only aggregate domains. It uses the AO abbreviation for AGGREGATEONLY.

```
APP ATM, DE, ST,AO
```

This examples extracts all ATM records for the last 1000 samples into a separate file named \$DATA.STORE.ATMFILE while suppressing terminal output.

```
APP ATM, EXTRACT $DATA.STORE.ATMFILE, NOPRINT, SAMPLES 1000
```

This example illustrates writing comma-separated value (CSV) records to a pre-existing entry-sequenced file named ATMCSV. Only aggregate ATM records are written to the CSV file, for 1000 samples. The TAB option replaces normal tab characters in the output with a comma.

```
APP/OUT ATMCSV/ATM, DE, ST, AO,RAW, TAB ",", NOLINEBREAK, S1000
```

The next example illustrates the EDL DETAIL command for a user-defined ASAP entity where the entity is controlled by a SET AGGREGATE ONLY command in ASAPXCNF. When SET AGGREGATE ONLY is defined for an entity, detail records are not written to the ASAP database, but they are available from ASAP memory using the ME (MEMORY) option. Setting this command in user EDL files lets you right-mouse on an object in ASAP Client and then select Show Object Details to show the detail records that make up the aggregate domain. The MI (MINSTATE) option is used to control record selection to choose only records where at least one attribute state is equal to or greater than the low-alert state (3), and the CO (COUNT) parameter is

used to limit the number of lines of response to 20. Setting this command lets you immediately retrieve the alerting detail records from an aggregate domain where only the aggregate domain is written to the ASAP database and to the ASAP Client. Option abbreviations are used in the EDL DETAIL command because the domain name insertion (^) can insert up to 64 additional bytes into the command. Completely

CIP Command

```
CIP [/OUT <file>/] [\* ] [ [.] * ] [,SAMPLES count ]
                    [\node] [ [.] domain][,STATES ]
                    [,TIME hh[:mm [m/d/y]]]
                    [, MINSTATE [AUTO] ]
                    [ [state] ]
                    [, COUNT count ]
                    [,EXTRACT filename ]
                    [,NOPRINT ]
                    [,NOLINEBREAK ]
                    [,AGE minutes ]

    entity [\* ] [ [.] * ] [, DETAIL ]
          [\node] [ [.] domain][,SAMPLES count ]
          [,STATES ]
          [,TIME hh[:mm [m/d/y]]]
          [, MINSTATE [AUTO] ]
          [ [state] ]
          [, COUNT count ]
          [,EXTRACT filename ]
          [,NOPRINT ]
          [,NOLINEBREAK ]
          [,AGE minutes ]
```

The CIP command displays CIP entity statistics for the specified nodes, entities, and domain names.

entity Any CIP sub-entity name. Valid sub-entity names are:

CIPCLIM	(or CLIM)
CIPMONITOR	(or MONITOR)
CIPPROVIDER	(or PROVIDER)

domain

Any CIP domain (object) name

SAMPLES

the number of samples to display per domain

STATES

the objective states of CIP metric values

TIME

defines ending time and date of sample(s) to display

DETAIL

displays custom detailed CIP metric values for specific sub-entities

MINSTATE

displays records based on attribute states

state

show records with at least one attribute state equal to or higher than *<state>*. Valid states are 1 through 8

AUTO

show records with at least one attribute state greater than state 2 (OK)

COUNT

defines the max number of records to display

EXTRACT

extract CIP records from the ASAP database to a separate database file or to a process. If the specified file does not exist, it will be created

NOPRINT

suppress terminal output for EXTRACT commands

NOLINEBREAK

do not break output lines at 132 characters. Be sure the output file can handle lines longer than 132 bytes before using this option

AGE

return removed objects for the past number of minutes specified

Examples

- To show the latest CIP statistics for the default node:
`CIP`
- To show the latest CIP statistics for all nodes:
`CIP *`
- To show the latest CIP CLIM detailed statistics for the local node:
`CIP CLIM, DETAIL`
- To show the latest detailed CIP Monitor statistics for Zcam0 on \Chicago:
`CIP MONITOR \CHICAGO.ZCAM0, DETAIL`
- To show the latest detailed CIP Provider statistics for the last four samples:
`CIP PROVIDER, DE, S4`
- To extract the latest CLIM statistics for all CLIMs for 1000 samples to a file:
`CIP CLIM, EXTRACT $DATA.DATA.CLIM, NOPRINT, SAMPLES 1000`
- To extract the last 1000 samples of CLIM statistics into a CSV file:
`CIP/OUT CLIMCSV/CLIM, RAW, DE, ST, TAB ",", NOLINEBREAK,
SAMPLES 1000`

CLEANUP Command

The CLEANUP command invokes the database cleanup function. How the cleanup function works depends on the RETAIN option assigned to the Collect server.

Caution

This command purges or releases statistics. This command can cause degradation in performance when the RETAIN option is set to a value greater than one. Only advanced ASAP users should use this command.

Type `STATUS COLLECT` to display the RETAIN option setting of a Collect process. The information is the last line of output in parenthesis.

CLEANUP [\node] COLLECT | collect-pid

node

specifies the individual system for which information is to be displayed.

collect-pid

specifies a Collect process ID.

Example

This example depicts a cleanup based on the RETAIN value for the Collect process:

```
+ CLEANUP COLLECT          ! Do cleanup based on RETAIN value for the
! Collect process
```

COMM Command

The COMM command displays COMM entity availability statistics for the specified nodes and communication lines. For an explanation of COMM performance statistics, type HELP COMMSTATS. For an alphabetical list, see the [COMM Statistics](#) table.

```
COMM [/OUT file/] [\*      ]           [, SAMPLES count ]
                [\node ] [ [.] commline [, TIME hh [:mm [m/d/y ] ] ] ]
                                                         [, CPU number   ]
                                                         [, INFO        ]
                                                         [, PERCENT     ]
                                                         [, RATE       ]
                                                         [, STATE      ]
```

OUT file

specifies the spooler or line printer where the ASAP output is to be sent. When no output file is specified, the output is displayed on your terminal or workstation.

*

specifies that the performance and state information for all systems is to be displayed.

node

specifies the individual system for which statistical information is to be displayed. If you want information for all the systems, specify COMM *. If you do not specify a node or the wild-card character (*), the default is used. The default is the system

where the ASAP program was started, or the system specified in the ASAP SYSTEM command.

commline

specifies the individual communication line which performance and state information is to be displayed. All the comm lines from all the collected nodes can be output by specifying COMM *.*. If you do not specify COMM or the wild-card character (*), the default is used. The default is all the communication lines on the current system.

*

specifies that the performance and state information for all communication lines on the specified system is displayed. If you do not specify a communication line or the wild-card character (*), statistics for all communication lines on the current system are displayed.

SAMPLES *count*

specifies the number of sample periods to be displayed. A sample period is defined by the RATE option in the SET command. You can specify 0 through 255 sample periods. The default sample period number is 1.

TIME *hh:mm m/d/y*

defines the ending time of the display information in hours, minutes, month, day, and year. For example, specifying "TIME 10:15" when RATE is 5 indicates that statistics collected from 10:10 to 10:15 are displayed.

When a "?" appears immediately after the timestamp, it indicates that the statistics are questionable and that there was a problem obtaining the statistics. Possible reasons for questionable statistics include:

- Down communication line.
- Communication line starting, connecting, or in a special state for diagnostics.

When a "<" appears immediately after the timestamp, the statistics were received late by the Collector process. Late statistics indicate busy Expand lines.

hh

is the hour.

mm

is the minute.

m/d/y

is the month, day, and year. Valid m/d/y entries are mm/dd/yy, yyyy/mm/dd, or mm/dd/yyyy.

CPU *number*

lists the statistics for the communication line are currently operating in the specified CPU number, cpu-number.

INFO

lists the operational information for monitored communication lines.

DATE

displays the date of the sample converted to the database local civil time.

TIME

displays the time of the sample converted to the database local civil time.

STATE

displays the operational state of the device. A device could be in these states:

Up	Communication line is operational
Down	Communication line is down and no longer operational.
LnkDn	X25 link level is down
Aborting	G-series Communication line is going to the down state, and is not operational.
Testing	G-series Communication line is being tested and is not operational.
Starting	Communication line is coming back to an operational state.
Stopping	G-series Communication line is in the process of going to down state and is not operational.
Suspend	G-series Communication line is in the suspended state and is not operational.
Unknown	G-series Communication line is in an unknown state and is not operational.
Suspending	G-series Communication line is entering a suspended state and is not operational.
Serviced	G-series Communication line is being serviced and is not

operational.

Special D-series Communication line is in the special state and is not operational.

Exclusive D-series Communication line is in the exclusive state and is not operational.

LDEV

displays the logical device number of the communication line.

PIN

displays the process identification number of the primary process.

TYPE

displays the device type and the subtype of the communication line.

D.xx NonStop™ OS displays the following:

CTRL

displays the controller address (in Octal) that the device was using.

UNIT

displays the unit number of the communication line.

CH

displays the channel that the communication line was using.

PROGRAM

displays the communication lines object file name.

G.xx NonStop™ OS displays the following:

GRP

displays the group number that the device was using.

MOD

displays the module number of the communication line.

SLOT

displays the slot number of the communication line.

PROGRAM

displays the object file name of the communication lines.

PERCENT

is the default for the COMM command. This option displays communication utilization in requests per second, percentage busy, percentage reading, and percentage writing.

Retry and transactions are displayed as per second, and Response time is displayed as average response time. For a description of the performance statistics, see the [COMM Statistics](#) table.

RATE

displays communication line utilization in I/Os per second, including rate per second, reads per seconds, writes per second, input KB per second, output KB per second, input data KB per second, output data KB per second. For a description of the performance statistics, see the [COMM Statistics](#) table.

STATE

shows all attributes that have an associated state and their state values. The option STATE can be used with the options RATE and PERCENT (default).

COMM Statistics

The following table lists the COMM statistics and characteristic options. You can also specify HELP COMMSTATS to display a list of disk statistics and options.

Table 6-2. COMM Statistics and Descriptions

Statistics	Option/Description
BUSY	Percent communication busy for the sample. (reading, writing, retrying).
INDATA	Number of data kilobytes read per second from the communication line.
INKB	Number of kilobytes read per second from the communication line.
OUTDATA	Number of data kilobytes written per second from the communication line.
OUTKB	Number of kilobytes written per second to the communication line.
RATES	Total reads and writes per second.
READ	Number of reads per second.
READ%	Percent communication busy reading data.

RETRY	Number of I/O process retries because of communication line error.
REQUESTS	Number of comm I/O requests (reads, writes) received by the communication process for the sample.
RSPTIM	Time that the I/O process spent on terminal response for all measured subdevices on the communication line.
TIME	Time of the statistics converted to data base local civil time.
TXACTN	Number of terminal transactions per second performed by the I/O processes for all the subdevices on the communication line.
WRITE	Number of writes per second.
WRIT%	Percent communication busy writing data.

Examples

These examples show what you can display with the COMM command:

```
+COMM
\CENTDIV   Date   Time   Status   Requests  Busy  Read  Write  Retry  Txactn  RspTim
-----
$X25A      6/22 19:45  Up       0.35 29.9 23.7  2.2   0.0   0.0     0
$EL70      6/22 19:45  Up       0.43 54.3 19.3 35.0  0.0   0.0     0

+COMM, RATE
\CENTDIV   Date   Time   Rate     Reads  Writes  InKb   OutKb   InData  OutDat
-----
$X25A      6/22 19:54  0.30    0.1    0.2    7.0   9.2    4.1    3.
$EL70      6/22 19:54  0.73    0.5    0.2   10.3  11.5   8.6    7.
```

COMMIT Command

The COMMIT command instructs ASAP to reload objectives after the ASAP Objectives2 database has been modified. It sends COMMIT requests to all entities on all nodes with data that has been modified during the current ASAP CI session.

COMMIT

CPU Command

The CPU command displays CPU entity availability statistics for the specified nodes and CPU numbers. Type HELP CPUSTATS for an explanation of CPU performance statistics. For an alphabetical list, see the [CPU Statistics](#) table. CPU state information appears only when a CPU is down. The down state information appears under the output heading M Q Busy field as Down. Up CPUs always contain some performance statistics, even when a CPU is 0% Busy.

```

CPU  [/OUT <file>/] [ \* ] [ [.] * ] [ , SAMPLES <count> ]
      [ \<node> ] [ [.] <cpu> ] [ , DETAIL ]
                                      [ , PERCENT ]
                                      [ , USE ]
                                      [ , CONFIGURED ]
                                      [ , HISTO ]
                                      [ , AVG ]
                                      [ , TYPE ]
                                      [ , TIME hh[ :mm [m/d/y]] ]
                                      [ , STATE ]
                                      [ , MEMORY ]
                                      [ , MB ]

```

OUT file

specifies the spooler or line printer where the ASAP output is to be sent. When no output file is specified, the output is displayed on your terminal or workstation.

specifies that the performance and state information for all systems is to be displayed.

node

specifies the individual system for which statistical information is to be displayed. If you want information for all the systems, specify CPU *. If you do not specify a node or the wild-card character (*), the default is used. The default is the system where the ASAP program was started, or the system specified in the ASAP SYSTEM command.

cpu

specifies the individual CPU for which performance and state information is to be displayed. All of the CPUs from all of the collected nodes can be output by specifying CPU *.*. If you do not specify a CPU or the wild-card character (*), the default is used. The default is all the CPUs in the current system.

specifies that the performance and state information for all CPUs on the specified system are to be displayed. If you do specify a CPU or the wild-card character (*), statistics for all CPUs in the current system are displayed.

SAMPLES count

specifies the number of sample periods to be displayed. A sample period is defined by the RATE option in the SET command. You can specify 0 through 255 sample periods. The default sample period number is 1.

DETAIL

displays all statistics for the specified CPU or system. Displays high PIN PCB configuration and usage statistics for D-series and later systems. On C-series systems, the display heading is named LCB, on D-series systems and later, the display heading is named PCBX.

PERCENT

displays the percentage of the total resource values for Memory pages, Control blocks (PCB, PCBX, LCB, and TLE), System, and MapPool.

USE

displays usage statistics for Memory pages, Control blocks (PCB, PCBX, LCB, and TLE), System, and MapPool.

CONFIGURED

displays the configured values for Memory pages, Control blocks (PCB, PCBX, LCB, and TLE), System, and MapPool.

HISTO

displays a histogram of CPU Busy, CPU Queue Length, and Memory Queue length. This is the default for the CPU command. The output for the histogram field is labeled M,Q.Busy.

AVG

displays the statistical averages across all running CPUs for the specified system during each sample period. This is a one-line display per system.

STATE

shows all attributes that have an associated state.

TYPE

displays the CPU model. Some examples of the CPU types are CLX800, K20000, S7000, and S70000.

TIME *hh:mm m/d/y*

defines the ending time of the display information in hours, minutes, month, day, and year. For example, specifying `-TIME 10:15ll` when `RATE` is 5 indicates that statistics collected from 10:10 to 10:15 are to be displayed.

When a `-?!l` appears immediately after the timestamp, it indicates that the statistics are questionable and that there was a problem obtaining the statistics. Possible reasons for questionable statistics include:

- CPUs are down.
- A CPU is being reloaded.
- A CPU was configured but does not physically exist.

When a `-<ll` appears immediately after the timestamp, the statistics were received late by the Collector process. Late statistics are an indication of busy Expand lines.

hh

is the hour.

mm

is the minute.

m/d/y

is the month, day, and year. Valid *m/d/y* entries are *mm/dd/yy*, *yyyy/mm/dd*, or *mm/dd/yyyy*.

MEMORY

Displays additional memory statistics: Page size, Total memory size, Swappable pages, Locked pages, Free pages, and Max locked pages.

MB

Displays the enhanced memory values in megabytes.

RAW

displays USE statistics in a dense tabular format rather than the standard ASAP sparse format.

CPU Statistics

The following table lists the CPU statistics and options displayed when you enter the ASAP HELP CPUSTATS command.

Table 6-3. CPU Statistics and Descriptions

Statistics	Description/Option
BUSY	The average percentage CPU busy for the sample
CHIT	The disk cache hit rate for this CPU in hits per second
DISC	The disk I/O rate for the CPU in I/Os per second

DISP	The dispatch rate for this CPU in dispatches per second
ET	The elapsed time of the sample in minutes up to TIME
FREEPAGES	Free memory pages available
IBUSY	The average percentage CPU interrupt busy for the sample
LCB	The number of link control blocks
IPUs	The number of IPUs in the logical CPU
LOCKEDPGS	Virtual Memory locked pages
MAP	MapPool pages used for the sample
MAXLOCKED	Max amount of Virtual pages ever locked
MEMQ	The average memory manager queue length
MEMSIZE	Total physical memory size
MemSwap	Swappable memory
MemLock	Memory locked
MemFree	Free memory available
MemLockMax	Max amount of memory
MemLockHigh	Highest amount of locked memory since CPU loaded
PAGESZ	Number of bytes per page
PCB	The number of low process control blocks
PCBX	The number of high process control blocks
PERCENT	The control block usage expressed as a percentage
PGLK PGS	The number of physical memory pages currently locked
QLEN	The average CPU queue length (processes waiting to execute)
SBSY	The average percentage CPU send busy for the sample
SEG	The number of virtual memory segments (Dxx and later) for USE/Conf
SWAP	The memory manager page fault rate in page swaps per second
SWAPPABLE	Number of swappable pages
SYS	The number of Sys Pool words for USE/CONF
TIME	The time of the sample converted to database local civil time
TLE	The number of time list elements currently in use
USE/CONF	Options that control whether statistics are current usage or configured values

Table 6-4. CPU Statistics Favorable for System Performance

Statistics	Displays Only When You Specify the...
FREEPAGES	MEMORY or MB option. Free memory pages available.
LOCKEDPGS	MEMORY or MB option. Memory locked pages.

MAP	USE or CONFIGURED options.
PCB	PERCENT, USE, or CONFIGURED option.
PGS	PERCENT, USE, or CONFIGURED option.
SEG	USE or CONFIGURED option (Dxx and later). SEG is the number of virtual memory segments you have specified.
SWAPPABLE	MEMORY or MB option. Swappable is the number of swappable memory pages.
SYS	USE or CONFIGURED option (Cxx only).
TLE	PERCENT option.

Examples

These examples show some of the available options and their output:

```
+CPU \DOME,S4, TIME 7:45 02/26/1999 !Show last 4 samples for system
\Chicago
\DOME
```

	Time	M,Q.Busy_	Busy	Qlen	Disp	Disc	Chit	Swap	MemQ	PgLk	Ibsy	Sbsy
			%									
00	2/26	7:45	.	1	1	47	2	2		4127	.2	
		7:44	.	1	1	49	2	2		4126	.2	
		7:43	.	1	1	48	2	2		4127	.2	
		7:42	.	1	1	47	2	2		4126	.2	
01	2/26	7:45	.	1	1	34				3830	.2	
		7:44	.	1	1	35				3830	.2	
		7:43	.	1	1	34				3831	.2	
		7:42	.	1	1	34				3831	.2	
02	2/26	7:45?	Down	0								
		7:44?	Down	0								
		7:43?	Down	0								
		7:42?	Down	0								
03	2/26	7:45?	Down	0								
		7:44?	Down	0								
		7:43?	Down	0								
		7:42?	Down	0								

Note

In this example of the DEFAULT output, the state for CPUs 2 and 3 appear because these CPUs were down.

```
+CPU \DOME,CONF
\DOME  \245      Et  Bsy  QL  Disp  Disc  Cht  Swp  MQ  Pgs   Pcb  Pcbx  Tle  Seg  Map
-----
00 2/25  8:05    1   1   1  46   2    2           16384 255  544  600 32767
01 2/25  8:05    1   1   1  34           16384 255 1544  600 32767
02 2/25  8:05?   1   0
03 2/25  8:05?   1   0
```

In this example, the output for CPUs 2 and 3 has a ? after the time, indicating that the statistics are not valid. These CPUs were down.

```
+CPU \SOLAR,MB
\SOLAR  Time      PageSz  MemSize  MemSwap  MemLocked  MemFree  MemLckMax
-----
00  3/17 22:54    16384    4295    4294      542      3597      3757
01  3/17 22:54    16384    4295    4294      533      3607      3757
02  3/17 22:54    16384    4295    4294      356      3868      3757
03  3/17 22:54    16384    4295    4294      354      3877      3757
```

In this example the output displays the enhanced memory statistics in megabytes.

DB Command

The DB command displays information about ASAP data base file disk space utilization when the SET DBMAXDAYS *days* option is enabled. When the SET DBMAXDAYS option is enabled, ASAP automatically manages and archives database history. For more about DBMAXDAYS features, see SET DBMAXDAYS.

```
DB [ year          ]
   [ DETAIL        ]
   [ INFO          ]
   [ entity-name  ]
```

year

specifies the year. If specified, database information will be for the subvolume associated with the specified year. For example, if the SET DBRolloverSubvolPrefix is "ASAP", then the command, DB 2003, will display disk space usage stats for ASAP database files in the subvolume ASAP2003.

DISK Command

The DISK command displays DISK entity availability statistics for the specified nodes and Disk volumes. For an explanation of DISK performance statistics, type HELP DSKSTATS. For an alphabetical list, see the [DISK Statistics](#) table.

DISK	[/OUT <i>file</i> /]	[*]	[[.] *]	[, SAMPLES <i>count</i>]
	[<i>node</i>]	[[.] <i>volume</i>]		[, TIME <i>hh</i> : <i>mm</i> [<i>m/d/y</i>]]
				[, AVG]
				[, CPU <i>number</i>]
				[, INFO]
				[, PERCENT]
				[, QUEUE]
				[, RATE]
				[, STATE]

OUT file

specifies the spooler or line printer where the ASAP output is to be sent. When no output file is specified, the output is displayed on your terminal or workstation.

*

specifies that the performance and state information for all systems is to be displayed.

node

specifies the individual system for which statistical information is to be displayed. If you want information for all the systems, specify DISK *. If you do not specify a node or the wild-card character (*), the default is used. The default is the system where the ASAP program was started, or the system specified in the ASAP SYSTEM command.

volume

specifies the individual DISK volume for which performance and state information is to be displayed. All the disks from all the collected nodes can be output by specifying DISK *.*. If you do not specify a DISK or the wild-card character (*), the default is used. The default is all the DISK devices on the current system.

*

specifies that the performance and state information for all disks on the specified system is displayed. If you do not specify a disk volume or the wild-card character (*), statistics for all disks on the current system are displayed.

SAMPLES *count*

specifies the number of sample periods to be displayed. A sample period is defined by the RATE option in the SET command. You can specify 0 through 255 sample periods. The default sample period number is 1.

TIME *hh:mm m/d/y*

defines the ending time of the display information in hours, minutes, month, day, and year. For example, specifying -TIME 10:15|| when RATE is 5 indicates that statistics collected from 10:10 to 10:15 are displayed.

When a -?|| appears immediately after the timestamp, it indicates that the statistics are questionable and that there was a problem obtaining the statistics. Possible reasons for questionable statistics include:

- Down Disk Drives or down volume
- Disk being revived, formatted, or mounted

When a -<|| appears immediately after the timestamp, the statistics were received late by the Collector process. Late statistics indicate busy Expand lines.

hh

is the hour.

mm

is the minute.

m/d/y

is the month, day, and year. Valid *m/d/y* entries are *mm/dd/yy*, *yyyy/mm/dd*, or *mm/dd/yyyy*.

AVG

displays the largest statistical values for the entire volume. Only one line of output is displayed. This can be used in conjunction with the CPU, RATE, PERCENT, and QUEUE options.

CPU *number*

lists the statistics for disk devices that IOPs are currently operating in the specified CPU number, *cpu-number*.

INFO

lists the operational information for primary disk devices and the mirror devices if they exist.

DATE

displays the date of the sample converted to the database local civil time.

TIME

displays the time of the sample converted to the database local civil time.

LDEV

displays the logical device number of the disk volume.

CPCTYMB

displays the total formatted disk space in megabytes.

GBUSED

displays the total space used at the time of the sample in gigabytes.

Max ExtMB

displays the largest extent available for the volume.

MB AVAIL

displays the total space available at the time of the sample in megabytes.

ADDR

displays the controllers octal address that is controlling the disk device. D-series only.

UNIT

displays the unit number of the disk device. D-series only.

GRP

displays the group number. G-series.

MOD

displays the module number. G-series.

SLT

displays the slot number. G-series.

STATE

displays the operational state of the device. A device could be in these STATES:

Up

device is operational.

Down

entire device is down.

HDown

entire device is in a Hard down state.

-P HD

primary path is Hard down.

-P Dn

primary path is Down.

-B H

backup path is Hard down.

-B Dn

backup path is Down.

-M H

mirror path is Hard down.

-M Dn

mirror path is Down.

-MB H

mirror backup path is Hard down.

-MB D

mirror backup path is Down.

Reviv

device is being revived.

Frmtg

device is being formatted.

Spec1

device is in the Special state.

ERROR

displays the Error reason when the disk statistics are invalid.

CpuDn

IOP switched to another CPU. The sample for the devices is invalid for one sample interval.

Cpusw

IOP for the disk volume switched to another CPU. The sample for the devices is invalid for one sample.

DctCh

new disk volume discovered. The sample is invalid for all disk devices for one sample interval.

NoCtr

unable to obtain DP2 counters. Check the ASAP log for additional error messages. Performance statistics are unavailable. State information is available.

NoMes

unable to communicate with Measure subsystem. For more error messages, check the ASAP log. Performance statistics are unavailable. State information is available.

Pthsw

controller path switch occurred. The sample for the device is invalid for one sample interval.

SteCh

device state change. The sample for the device is invalid for one sample interval.

UnAvl

device configured but the disk process for the device has not started.

PERCENT

is the default for the DISK command. This option displays disk utilization in percentage busy, percentage reading, percentage used, and percentage writing. Chit, Queue length, and Swaps are displayed as per second. For a description of the performance statistics, see the [DISK Statistics](#) table.

QUEUE

displays disk queues lengths of read or write operations that are waiting to execute. The statistics displayed are QLEN, QM, VQLN, VM, RBLK, INPUTKB, OUTPTKB, and BSPLT. For a description of these statistics, see the [DISK Statistics](#) table.

RATE

displays disk utilization in I/Os per second, including: rate per second, reads per seconds, writes per second, Chit, Queue length, and Swaps per second. GBUSED shows the number of gigabytes allocated. For a description of the performance statistics, see the [DISK Statistics](#) table.

STATE

shows all attributes that have an associated state. Associated states are displayed for the disk command without any options, and the disk command with these options: RATE, QUEUE, AVG, PERCENT (Default).

DISK Statistics

The following table lists the DISK statistics and characteristic options. You can also specify HELP DISKSTATS to display a list of disk statistics and options.

Table 6-5. DISK Statistics and Descriptions

Statistics	Option/Description
BSPLT	The block split total of 512 through 4096 block splits for the volume.
BUSY%	Total percentage disk busy during read, write, and seek operations.
CHIT	Cache hits. Total of 512 through 4096 blocks that occurred per second.

DATE	Date of the sample converted to the database local civil time.
INPUTKB	The number of kilobytes read per second from the disk.
GBUSED	Total gigabytes of disk space allocated.
Max ExtMB	The largest free extent available for the volume.
MBUSE%	Percentage of disk space allocated.
MB AVAIL	Total megabytes of disk space allocated.
OUTPTKB	The number of kilobytes written per second to the disk.
QLEN	The disk queue length of read/write operations waiting to execute.
QM	The maximum number of items in the queue described by QLEN.
RATE	The total utilization of reads, writes, and seeks per second.
RBLK	The number of operations blocked because of file or record locks.
READ	Number of reads per second.
READ%	Percentage of time spent reading.
REQUESTS	The number of I/O requests per second received by the disk process.
STATE	State of the drive; the DISK INFO command for the list of possible states.
SWAP	Number of memory page swaps per second to or from disk.
VQLN	The disk volume semaphore queue length (exclusive and shared modes operation).
VQ	The maximum number of volume semaphore requests for the semaphore queue.
WRITE	Number of writes per second.
WRITE%	The percentage of time spent writing.
TIME	The time of the sample converted to database local civil time.

Examples

These examples show what you can display with the DISK command:

```
+DISK, QUEUE
\CENTDIV  Date Time  Qlen Qm  VQln Vm Requests ReqBlk  InputKB  OutptKB BSplt
-----
$BR-P    4/28 14:50      2      1      0.00      0      0
$BR-M           14:50      2      1      0.00      0      0
$CO-P    4/28 14:50      1      0      0.00      0      0
$CO-M           14:50      1      0      0.00      0      0
$DSV-P   4/28 14:50      1      0      0.00      0      0
$MC-P    4/28 14:50      2      3      0.10      0      0
$MC-M           14:50      2      3      0.10      0      0
$MM-P    4/28 14:50    .93 3  1.72 3  100.80    40     37
$MM-M           14:50    .93 3  1.72 3  100.80    35     37
$RTOOL-P 4/28 14:50      1      3      0.00      0      0
```

```

$SYSTEM-P 4/28 14:50      3      3      1.20      0      0
$SYSTEM-M      14:50      3      3      1.20      0      0

+DISK, RATE
\CHITOWN Date Time State GBUUsed Requests Rate Writes Reads Swap Qlen
-----
$DSV-P 4/29 13:25 Up 1.57 0.00
$NSX-P 4/29 13:25 Up 0.58 23.00 23.5 11.3 12.2 .93
$ROJO-P 4/29 13:25 Up 0.54 0.00
$ROJO-M      13:25 Up 0.54 0.00
$SYSTEM-P 4/29 13:25 Up 1.22 0.80
$SYSTEM-M      13:25 Up 1.22 0.80

```

EDL Command

The EDL command performs a syntax-only check of the specified EDL file.

EDL <i>filename</i> [STATS SUMMARY LIST DUMP COMMENT SUPPRESS]
--

STATS

shows syntax check statistics and error summary of EDL file.

SUMMARY

shows summary of entity and attribute names in EDL file.

LIST

shows listing of all statements in EDL file.

DUMP

lists and dumps each EDL control block; for debugging only.

COMMENT

prefixes compile info with -- comment characters (used by Client).

SUPPRESS

suppresses everything except errors in EDL file.

Examples

```
EDL $System.System.AsapSys2, LIST
```

```
EDL $System.System.AsapSys2, SUMMARY
```

EXIT Command

The EXIT command stops the Conversational Interface.

```
+EXIT
```

FC Command

The FC command retrieves and displays the last ASAP command you entered so you can modify and re-execute it. The FC command uses the subcommands R, I, and D to replace, insert, or delete characters. For more information about this command, see the Guardian User's Guide.

```
+FC
```

FILE Command

The FILE command displays FILE entity statistics and operational state information for the specified nodes and file domains. For an explanation of FILE statistics, enter HELP FILESTATS at an ASAP prompt.

```
FILE [/OUT file/] [ \* ] [, SAMPLES count ]
      [node ] [ [.] domain ] [, TIME hh[:mm [m/d/y] ]
                                [, STATES ]
                                [, DETAIL ]
                                [, MINSTATE [state ]
                                    [AUTO ] ]
                                [, EXACT ]
                                [, COUNT max-value ]
                                [, AGGREGATE ]
                                [, AGGONLY | AO ]
                                [, MEMORY ]
                                [, NOAGG ]
                                [, MINDB ]
```

OUT file

specifies the spooler or line printer where the ASAP output is sent. When no output file is specified, the output is displayed on your terminal or workstation.

*

specifies that the performance and state information for all systems is to be displayed.

node

specifies the individual system for which statistical information is displayed. If you want information for all the systems, specify FILE *. If you do not specify a node or the wild-card character (*), the default is used. The default is the system where the ASAP program was started or the system specified in the ASAP SYSTEM command.

domain

specifies the individual file domains for which performance and state information is displayed. All the file domains from all the collected nodes can be output by specifying FILE *. If you do not specify a file domain or the wild-card character (*), the default is used. The default is all the file domains on the current system.

*

specifies to display the performance and state information for all monitored file domains on the specified system. If you do not specify a file or the wild-card character (*), statistics for all files on the current system are displayed.

SAMPLES *count*

specifies the number of sample periods to be displayed. A sample period is defined by the RATE option in the SET command. You can specify 0 through 255 sample periods. The default sample period number is 1.

TIME *hh:mm m/d/y*

defines the ending time of the display information in hours, minutes, month, day, and year. For example, specifying -TIME 10:15|| when RATE is 5 indicates that statistics collected from 10:10 to 10:15 are displayed.

When a -?|| appears immediately after the timestamp, it indicates that the statistics are questionable and that there was a problem obtaining the statistics. The Error attribute usually indicates the cause of the problem.

When a -<|| appears immediately after the timestamp, the statistics were received late by the Collector process. Late statistics indicate busy Expand lines or nodes not synchronized in the network.

hh

is the hour.

mm

is the minute.

m/d/y

is the month, day, and year. Valid *m/d/y* entries are *mm/dd/yy*, *yyyy/mm/dd*, or *mm/dd/yyyy*.

STATES

shows all attributes that have an associated state.

DETAIL

shows all available statistics attributes, without states.

MINSTATE

displays files from memory in a subvolume or aggregate domain, depending on the selected option:

state

shows files with an OEM-state greater than or equal to *state*.

AUTO

shows files with an OEM-state higher than 2, or else shows all files.

EXACT

shows statistics for only the specified *domain*.

COUNT *max-value*

indicates the maximum number of subdomains to display.

AGGREGATE

shows statistics for aggregate domains. AGGREGATE is on by default.

AGGREGATEONLY | AGGONLY | AO

shows only statistics for aggregate domains.

MEMORY

shows statistics for domains directly from ASAP memory. Use the MEMORY option to view the individual domains that make up an AGGREGATEONLY domain.

NOAGG

suppresses the display of aggregate domains.

MINDB

filters display records from a historical database based on the supplied value state.

Considerations

- You cannot use TIME, SAMPLES, or * with MINSTATE.
- You can only use EXACT and COUNT with MINSTATE.

Examples

For descriptions of the fields displayed in FILE command output, enter `help filestats`.

- To display a list of file domains being monitored and a few attributes:

```
+file
\CENTDIV File Domains      Status      Date  Time  Full Files  Error
-----
$Mm.filework              Corrupt     9/19  9:59  100      25    0
$Mm.xx.disk               Up          9/19  9:59   0        -    0
$System.system.asap       Up          9/19  9:59  49        -    0
+
```

- To display only the attributes associated with a state:

```
+file, states
\CENTDIV File Domains      Status      S Date  Time  Full S      Eof S
-----
$Mm.filework              Corrupt     7 9/19 10:00 100 7      978944 1
$Mm.xx.disk               Up          2 9/19 10:00  0 2        0 1
$System.system.asap       Up          2 9/19 10:00 31 2      1569704 1
```

continued...

```
Owner  S Rwp S Open S TMF S Files S Error S
-----
  0,0  1 aaaa 1 No  1 No  1      25 1    0 1
255,255 1 ncnc 1 No  1 No  1      - 1    0 1
255,255 1 nunu 1 Yes 1 No  1      - 1    0 1
+
```

To display all available attributes without the states:

```
+file, detail
\CENTDIV File Domains      Status      Date   Time   Error Full      Eof
-----
$Mm.filework               Corrupt     9/19 10:00    0 100    978944
$Mm.xx.disk                Up          9/19 10:00    0  0      0
$System.system.asap        Up          9/19 10:00    0  49    1569704
```

continued...

```
Owner   Rwep Open Tmf Files   Code Format Type      Parts LastOpen
-----
  0,0   aaaa No   No    25     -  -   -          -  -
255,255 ncnc No   No     -     0  1   Unstruct    0 2001/01/26 17:14
255,255 nunu Yes No     -    100  1   Unstruct    0 2001/03/14 08:45
```

continued...

LastModif

```
-----
-
2000/09/08 07:03
2001/03/08 13:57
+
```

To display all available attributes with the associated states:

```
+file, states, detail
\CENTDIV File Domains      Status      S Date   Time   Error S Full S
-----
$Mm.filework               Corrupt     7  9/19 10:00    0 1 100 7
$Mm.xx.disk                Up          2  9/19 10:00    0 1  0 2
$System.system.asapmon     Up          2  9/19 10:00    0 1  49 2
```

continued...

```
      Eof Owner   Rwep Open Tmf S Files S Code Format Type      Parts
-----
  978944  0,0   aaaa No   No    1    25 1   -  -   -          -
      0 255,255 ncnc No   No    1    - 1   0  1   Unstruct    0
 1569704 255,255 nunu Yes No    1    - 1  100  1   Unstruct    0
```

continued...

LastOpen

LastModif

```
-----
-
2001/01/26 17:14 2000/09/08 07:03
2001/03/08 13:57 2001/03/08 13:57
+
```

To display files in error in the subvolume domain \$mm.filework; that is, with at least one attribute with a state value greater than 2:


```
+file $mm.filework, states, minstate auto
\CENTDIV File Domains      Status      S Date   Time   Full S      Eof S
-----
$Mm.filework.filecorr     Corrupt     7  9/19 10:00   32 2      978944 1
$Mm.filework.filefull     Up          2  9/19 10:00  100 7      32768 1
continued...
```

```
Owner   S Rwp S Open S TMF S Files S Error S
-----
215,5   1 nunu 1 No   1 No   1     - 1     - 1
215,5   1 nunu 1 No   1 No   1     - 1     - 1
```

- To display only aggregate domains:

```
FILE, AGGONLY
```

- To configure ASAP Client to retrieve individual domain information directly from ASAP memory for aggregate-only domains when "Show Object Details" is selected, edit or add the FILE entity definition of this DETAIL statement, making sure to include the MIN, EX, COUNT and MEM options, and to use abbreviations as shown:

```
Detail      "FILE ^,ST,TAB,MIN AUTO,EX,COUNT 50,MEM"
```

GOAL Command

The GOAL command controls the objectives used to produce alerts and spawn actions within ASAP. It controls the set of objectives and gets information about those objectives.

```

GOAL [/OUT file/] [entity | *] [\node], COMMIT
        [. | ]][domain]], simplex-goal [event] [action]
        , Boolean-goal [event] [action]
        ,LIKE domain
        ,ACTIVATE
        ,DEACTIVATE
        ,DELETE [,SIMPLEX | GOALNAMES ]
        ,INFO [,OBEYFORM ]
        ,OBEYFORM
        ,LIST

simplex-goal attr op [value] [, simplex-goal...]

Boolean-goal gname=(attr op value [ AND|OR attr op value ... ] )
                [ AND|OR Boolean-goal ] )

                gname , DELETE

event          INFO|CRITICAL [REPEAT] [NODISPLAY] [down] [-up]
                NOEMS

action         ACTION "string" | rule [WAIT wait] [NOREPEAT]

```

Use GOAL command to set or view preferred-property-threshold settings (objectives). When a goaled property threshold is exceeded, the state of the goaled property is assigned an alert state. States in ASAP describe the status of a property: up, down, warning, or critical. ASAP displays goaled alert states in the ASAP Client, the Event Management Service (EMS) subsystem, or both.

To enable goaled property alerts in ASAP Client, set State Determination to UseStateGraphState for each goaled property in the ASAP Client Attribute Property sheet. For more info, see "Configuring ASAP" in ASAP Client help.

To enable goaled property events in the EMS subsystem, issue a SET OBJECTIVESEVENT ON command from the ASAP command interpreter before starting ASAP Server components.

The GOAL command controls objectives used to produce alerts and to take actions. The MONITOR command controls which objects are monitored within ASAP.

The GOAL command is also used to display objective values and actions.

The RANK command is a synonym for the GOAL command; the two are interchangeable.

entity

is the name of an ASAP Entity as defined in EDL.

node

is the node name the command applies to.

domain

is any form of an ASAP domain name. Domain names can represent physical devices (\$Data) or abstract representations of applications (Funds\Transfer\Input.) Some command options support the asterisk wild-card character at the end of the domain name.

COMMIT

sends a commit request to all entities or a specific entity on a local or remote node. This request causes ASAP to reload its set of objectives for the entity or entities on the node.

simplex-goal attr op [value] [, simplex-goal ...]

*Boolean-goal gname=(attr op value [AND|OR attr op value...])
[AND|OR Boolean-goal])
gname, DELETE*

()

Parentheses are required for Boolean goals. They are used to group one or more attribute-operator-value sets. Parentheses can be nested as needed. Boolean goals are processed left to right from the innermost parentheses outward.

attr

is the name of an attribute name as defined in ASAP EDL for the entity. The attribute must be defined with StatePair Yes, and it cannot be the StateIsOp (STATUS) attribute. Note to specify an event or action on the StateIsOp attribute, specify STATUS or STATUS NOEMS instead of the <attr> attribute name.

op

is one of the following objective operators:

- > The attribute value must be greater than the objective value.
- >> The attribute value must be greater than the objective value. Used when defining more than one greater-than goal (a de-escalating goal.) Be sure to set ObjectivesEventState to HIGH or lower when using this operator.
- >>> The attribute value must be greater than the objective value. Used when defining more than one greater-than goal (a de-escalating goal.) Be sure to set ObjectivesEventState to HIGH or lower when using this operator.

- < The attribute value must be less than the objective value.
- << The attribute value must be less than the objective value. Used when defining more than one less-than goal (an escalating goal.) Be sure to set ObjectivesEventState to HIGH or lower when using this operator.
- <<< The attribute value must be less than the objective value. Used when defining more than one less-than goal (an escalating goal.) Be sure to set ObjectivesEventState to HIGH or lower when using this operator.
- = The attribute value must equal the objective value.
- <> The attribute value cannot equal the objective value.
- >= The attribute value must be greater than or equal to the objective value.
- <= The attribute value must be less than or equal to the objective value.

op has no meaning for the STATUS attribute.

value

is the stated goal for this attribute. If no value is specified, the existing value for attribute-operator is deleted. The value must match the TypeData setting as defined for the attribute in EDL. (See SHOW command.) For example, you cannot set a decimal value objective for an attribute defined as an integer.

value has no meaning for the STATUS attribute. *gname*

is the name of a Boolean goal. The goal name can be from 1-32 bytes long. Boolean goals can be defined at the <entity> and the <domain> level. At a minimum, <entity> must be specified when defining a Boolean goal. The goal name must be followed by an equal sign and the Boolean goal itself must be enclosed in parentheses. A goal name specified at the <domain> level overrides a goal with the same name specified at the <entity> level.

= (*equal sign*)

The equal sign assignment is required for defining Boolean goals. It must immediately follow the goal name in a Boolean goal specification. The equal sign cannot be used for other goal specifications.

AND

is a logical operator used to combine attr-op-value sets into Boolean goals. When AND is specified it states that both sets must be true, failure of either attr-op-value comparison will cause the AND condition to fail.

OR

Is a logical operator used to combine attr-op-value sets into Boolean goals. When OR is specified it states that either set can be true for the goal to be satisfied, failure of both attr-op-value comparisons is required to cause the OR condition to fail.

event

INFO | CRITICAL [REPEAT] [NODISPLAY] [*down*] [*-up*]

INFO

configures ASAP to generate an informational event when the objective is not met.

CRITICAL

configures ASAP to generate a critical event when the objective is not met.

REPEAT

causes ASAP to repeat the event for each interval where the objective is not met. REPEAT is always on for Boolean goals.

NODISPLAY

overrides any global settings to suppress display of the event on the operator console.

down

is the event number to use for the down event, when the objective comparison fails. Events must be numbered in the range 6000 through 9999. Event number 4000 is used when this is not specified. The ASAP EMS event template for the down event is not applied when custom down events are defined.

-up

is the event number to use for the up event, which is triggered when the objective comparison returns to normal after a failure. The number specified must be in the range 6000 through 9999. Event number 4001 is used when this is not specified. The minus sign must precede the up event number even if the down event number is not specified. The ASAP EMS event template for the up event is not applied when custom up events are defined.

For escalating and de-escalating goals the up event is written when the alert level returns to normal.

Up event specifications are not supported for Boolean goals.

STATUS

When events are turned on (using SET OBJECTIVESEVENT ON), status changes in ASAP system entities default to generate one critical EMS event. The STATUS attribute specifies changes to the default behavior for system entities. Use STATUS with ASAP user-defined entities to define all event behavior for status changes.

NOEMS

Used only with the STATUS attribute. NOEMS turns off automatic event generation for status changes in ASAP system entities.

action

```
ACTION string | rule [WAIT wait] [NOREPEAT]
```

string

is an action string where the first word in the string is a command interpreter name such as TACL, SCF, or FUP, and the remainder of the string represents the command to be executed by that command interpreter.

The string must be enclosed in double quotation marks, and ASAP converts any single quotes found within a string to double quotes. For example, ACTION "FUP SECURE <#OBJECT>, '<#GOAL>' " becomes FUP SECURE <#OBJECT>, "<#GOAL>" when processed by the ASAP Action server.

Whenever these tokens are contained in a string, the tokens in the string are replaced with values associated with the objective, and the resulting command is executed:

<#ENTITY> The entity name

<#OBJECT>	The NonStop™ OS or OSS object name
<#DOMAIN>	The domain name
<#D1>	Leftmost part of a multilevel domain name
<#D2>	2nd part of a multilevel domain name
<#D3>	3rd part of a multilevel domain name
<#D4>	4th part of a multilevel domain name
<#D5>	5th part of a multilevel domain name
<#ATTR>	The attribute name
<#OP>	The binomial objective operator. Operators are defined using the following 2-byte values: GT greater than G2 level 2 greater than G3 level 3 greater than LT less than L2 level 2 less than L3 level 3 less than EQ equal NE not equal GE greater than or equal to LE less than or equal to
<#VALUE>	The current value of the attribute
<#STATE>	The current state of the attribute
<#LASTSTATE>	The last state of the attribute
<#GOAL>	The objective value
<#GNAME>	The name of the Boolean goal
<#GOALVAL>	Attribute, operator, goal and value in a string with no spaces where goal and value are separated by the character. For example, "PRI=150 180". "PRI = 150" is the goal setting and 180 is the current value of the PRI attribute.
<#attrGOAL>	A specific goal token for the attribute. For example if the attribute name is BUSY the name of this token will be <#BUSYGOAL> and it will contain the goal for the BUSY attribute.
<#attrOP>	A specific operator token for the attribute. For example if the attribute name is BUSY the name of this token will be <#BUSYOP> and it will contain the operator used with the BUSY attribute.

<#attrVALUE> A specific value token for the attribute. For example if the attribute name is BUSY the name of this token will be <#BUSYVALUE> and it will contain the value of the BUSY attribute.

When using Boolean goals where one or more attribute-operator-value settings are defined, the values returned for the <#ATTR>, <#OP>, <#VALUE>, <#GOAL> and <#GOALVAL> are returned in a list separated by vertical bars. For example consider this GOAL statement:

```
GOAL CPU CPULOAD = (BUSY < 80 AND QUEUE < 2)
```

If CPU BUSY is 90 and the QUEUE length is 3, these values will be returned for the specified tokens:

```
<#ATTR>      BUSY|QUEUE
<#OP>        LT|LT
<#VALUE>     90|3
<#GOAL>      80|2
<#GOALVAL>   BUSY<80|90|QUEUE<2|3
```

The order of the items is dependent on the evaluation order of the goal statement, which is innermost from left to right.

If an attribute is used more than once in a Boolean goal statement then the names for attribute-specific tokens like <#BUSYGOAL> will use numbers immediately following the attribute portion of the name to uniquely identify the additional tokens. For example the second <#BUSYGOAL> token will be named <#BUSY2GOAL>.

rule

is an action rule defined using the ACTION command. Action rules are stored action strings that can be applied to multiple domains.

WAIT *wait*

indicates the number of minutes ASAP should wait before retrying an action string against a particular entity and domain. The wait time also

depends on the ASAP rate setting because ASAP validates objectives only once per cycle.

NOREPEAT

indicates that ASAP should not continue to repeat the action until the condition is corrected.

LIKE

sets attributes in the new or modified domain that are currently active in the LIKE domain. Any goal specified in the command then overrides corresponding LIKE domain values.

ACTIVATE

turns ranking on for the specified domain(s) in the specified entity. *entity* must be specified when domain is specified; otherwise both are optional. Domain names can use the asterisk wild-card character as the last character in the name.

Note

A double asterisk is required to select multiple File and Process records.

DEACTIVATE

turns ranking off for the specified domain(s) in the specified entity. *entity* must be specified when domain is specified, otherwise both are optional. Domain names can use the asterisk wildcard character as the last character in the name.

DELETE

deletes records from the database. *entity* must be specified when domain is specified; otherwise both are optional. Domain names can use the asterisk wild-card character as the last character in the name. DELETE using a wild-card domain name requires the command to be entered twice for confirmation before any action is taken.

SIMPLEX

deletes the simplex goals for an entity or domain, leaving any Boolean goals defined for the entity or domain.

GOALNAMES

deletes the Boolean goals for an entity or domain, leaving any simplex goals defined for the entity or domain.

INFO

returns information about objectives from the database. *entity* must be specified when domain is specified; otherwise both are optional. Domain names can use the asterisk wild-card character as the last character in the name.

OBEYFORM

returns objective information in OBEY file format suitable for playback to ASAP CI. *entity* must be specified when domain is specified; otherwise both are optional. Domain names can use the asterisk wild-card character as the last character in the name.

LIST

retrieves a list from the active set of objectives for all entities or for a specific entity. The LIST command communicates directly with running ASAP components to retrieve objective information. *entity* must be specified when domain is specified; otherwise both are optional. Domain names can use the asterisk wild-card character as the last character in the name.

Considerations

Since TCP/IP and Telserv entity names are long and difficult to type ASAP supports the following abbreviations for the entity names:

TCPARP	ARP
TCPICMP	ICMP
TCPICMP6	ICMP6
TPCIGMP	IGMP
TCPIP	IP
TCPIP6	IP6

TCPLOOP	LOOP
TCPMONGQ	MONGQ
TCPPORT	PORT
TCPPROCESS	TCP
TCPQIO	QIO
TCPROUTE	ROUTE
TCPRTE	RTE
TCPSOCK	SOCK
TCPSUBNET	SUBNET
TCPUDP	UDP
TELSERVPROCESS	TELSERV
TELSERVSERVICE	SERVICE
TELSERVWINDOW	WINDOW

Since CIP entity names are long and difficult to type ASAP supports the following abbreviations for the entity names:

CIPCLIM	CLIM
CIPMONITOR	MONITOR
CIPPROVIDER	PROVIDER

Examples

- To add or modify the record for the disk \$DATA:
GOAL DISK \$DATA, BUSY < 65
- To specify that a file must be less than 90 percent full:
GOAL FILE \$SYSTEM.FILE.FILE, FULL < 90
- To add or modify the CPU record, define Busy and Queue:
GOAL CPU 0, BUSY < 90, QUEUE < 4
- To set the objectives for a domain:
GOAL FUNDS TRANSFER, BUSY < 30, TRANSRATE >= 15, ERROR < 1

- To override global objectives and create critical events:
GOAL FUNDS TRANSFER\WEST, BUSY > 20, BUSY < 40 CRITICAL
REPEAT

- To turn ranking off for a file:
GOAL FILE \REMOTE.\$DATA.FILE.FILE, DEACTIVATE

- To commit FILE on \REMOTE:
GOAL FILE \REMOTE, COMMIT

- To save the disk objectives configuration:
GOAL /OUT FILE/ DISK, INFO, OBEYFORM

- To list a domain for ATM:
GOAL ATM\CHICAGO\ABC, LIST

- To display all records from the local database:
GOAL

- To deactivate a broken spooler device while it is repaired:
GOAL SPOOLER \$SPLG\DEV\SLP, DEACTIVATE

- To turn on all domains for all entities:
GOAL, ACTIVATE

- To commit all objectives changes:
GOAL, COMMIT

- To define an action to reload a CPU after it fails:
GOAL CPU, STATUS ACTION "TACL RELOAD <#DOMAIN>, PRIME" WAIT
15

- To define a goal and action to secure the NonStop™ OS userid file to "OOOO":
GOAL FILE \$SYSTEM.SYSTEM.USERID, RWEP = "OOOO" ACTION "FUP
SECURE <#DOMAIN>, '<#GOAL>' "

- To define a goal and action to compress a volume that is getting too full and to write a critical EMS event, but to perform the compression only once every two hours:

```
GOAL DISK, FULL < 90 CRITICAL ACT "DCOM <#DOMAIN>" WAIT 120
```

- To define custom up and down event numbers for CPU 0 getting too busy:

```
GOAL CPU 0, BUSY < 90 CRITICAL 4444-5555
```

- To define a custom down event number for CPU 0 getting too busy:

```
GOAL CPU 0, BUSY < 90 CRITICAL 4444
```

- To define a custom up event number for CPU 0 when it returns to normal:

```
GOAL CPU 0, BUSY < 90 CRITICAL -5555
```

- To define a goal for busy for an OSS shell process aggregate-only domain:

```
GOAL PROCESS /bin/sh\##, BUSY < 10
```

- To define a global goal and action to resecure any monitored file using the ASAP default RESECURE rule:

```
GOAL FILE, RWEP = "AOAO" ACTION RESECURE
```

- To define a goal that all foreign TCP/IP Port connections must originate from net 16:

```
GOAL TCPPORT, FADDR = 16.0.0.0
```

- To define a goal that all foreign TCP/IP Port connections on Port 23 must originate from subnet 16.150:

```
GOAL TCPPORT 23, FADDR = 16.150.0.0
```

- To define a goal to report the occurrence of any CIP Linux errors for any CLIM device:

```
GOAL CIPCLIM, AGTLNXERRS = 0, SVRLNXERRS = 0
```

- To define an escalating goal for the BUSY attribute for all CPUs:

```
GOAL CPU, BUSY < 50, BUSY << 70, BUSY <<< 90
```

- To define an escalating goal for the BUSY attribute for all CPUs using separate commands:

```
GOAL CPU, BUSY < 50
GOAL CPU, BUSY << 70
GOAL CPU, BUSY <<< 90
```

- To define a normal simplex goal on the CPU BUSY attribute:

```
GOAL CPU, BUSY << 80
```

Using the << or <<< operator is the same as using the < operator if only one goal is defined.

- To define a de-escalating goal on the TMF TRANSRATE attribute:

```
GOAL TMF, TRANSRATE >30, TRANSRATE >> 20, TRANSRATE >>> 10
```

- To define an escalating goal on CPU BUSY that generates an information EMS event and then a critical EMS event as the problem worsens:

```
GOAL CPU, BUSY < 50, BUSY << 70 INFO, BUSY <<< 90 CRITICAL
```

The -UPII event, if enabled, will be written when the CPU leaves the alert state by falling back below 50% busy.

- To define an escalating goal on CPU BUSY that generates specific event numbers and where the most critical event repeats while that level of problem exists:

```
GOAL CPU, BUSY < 50 INFO 8881-8880
GOAL CPU, BUSY << 70 CRITICAL 8882
GOAL CPU, BUSY <<< 90 CRITICAL REPEAT 8883
```

If the CPU is 70% busy when it alerts then event 8882 will be written.

- To define an escalating goal on the SPOOLER FULLPCT attribute that takes a different action depending on how full the SPOOLER is:

```
GOAL SPOOLER, FULLPCT < 50 ACTION "TACL CLEANSPL <#OBJECT>"
GOAL SPOOLER, FULLPCT << 70 ACTION "TACL DELETEJOB <#OBJECT>"
GOAL SPOOLER, FULLPCT <<< 90 ACTION "TACL FORCESPL <#OBJECT>"
```

- To define a simplex goal for all CPUs and override that goal with an escalating goal for CPU 0:

```
GOAL CPU, BUSY < 60
GOAL CPU 0, BUSY < 70, BUSY << 80, BUSY <<< 90
```

- To define a Boolean goal named BUSYGOAL for all CPUs that checks the BUSY and QUEUE attributes and where failure of either will result in the goal failing:

```
GOAL CPU, BUSYGOAL = (BUSY < 80 AND QUEUE < 3)
```

- To define a Boolean goal named BUSYGOAL for all CPUs that checks the BUSY and QUEUE attributes and where failure of both are required for the goal to fail:

```
GOAL CPU, BUSYGOAL = (BUSY < 80 OR QUEUE < 3)
```

- To define a Boolean goal named BUSYGOAL for all CPUs that checks the BUSY and QUEUE attributes and writes a critical event if the goal fails:

```
GOAL CPU, BUSYGOAL = (BUSY < 80 AND QUEUE < 3) CRITICAL
```

- To define a Boolean goal named BUSYGOAL for all CPUs that checks the BUSY and QUEUE attributes and takes an action if the goal fails:

```
GOAL CPU, BUSYGOAL = (BUSY < 80 AND QUEUE < 3) ACTION "TACL
RUN $SYSTEM.MACROS.BUSYCPU <#OBJECT> <#BUSYVALUE>
<#QUEUEVALUE>"
```

- To define a Boolean goal named BUSYGOAL for all CPUs that checks the BUSY, QUEUE and DISP attributes:

```
GOAL CPU, BUSYGOAL=(BUSY < 80 AND QUEUE < 3 AND DISP < 1000)
```

- To define a Boolean goal named RESOURCE for all CPUs that checks the BUSY, QUEUE and two PCB attributes:

```
GOAL CPU, BUSYGOAL=(BUSY<80 AND QUEUE<3 AND (PCB>10 OR
PCBX>10))
```

HELP Command

The HELP command provides an explanation for the command or keyword that you specify.

HELP [<i>command</i> ALL ERROR SET <i>option</i>]

command

specifies the command or keyword to display detailed help for.

ALL

provides a summary of all commands.

ERROR

lists all of the Monitor and Collect error messages by message number, description, and parameters. For a description of these messages, including suggested recovery actions, see the ASAP Messages Manual.

SET *option*

provides a description of a specific SET option.

Example

```
ASAP - Availability Stats and Performance - T0402V02.00 (29JUN2001)
HELP [ <command> | ALL | ERROR | SET <option> ]
```

The HELP command provides an explanation for the specified command(s).

```
HELP ALL           provides a summary of commands.
HELP ERROR        provides description of monitor/collect error msgs.
HELP SET <option> provides description of a specific set option.
HELP <command>    provides detailed help for that keyword/command.
```

Examples

```
HELP
HELP STATUS
HELP SET DB
```

LH Command

The LH command collects Expand line-handler entity availability statistics for the specified node names and line names. The Expand line-handler services incoming and outgoing requests and responses to transfer data between processes. For an explanation of line-handler statistics, type HELP NETSTATS.

LH	[/OUT <i>file/</i>] [*] [[.] *]	[, INFO]
	[\node] [[.]line]	[, FRAMES]
		[, HISTO]
		[, IP]
		[, LINE]
		[, OPERATIONS]

[, PATH]
[, POOL]
[, RATES]
[, SAMPLES <i>count</i>]
[, STATE]
[, TIME <i>hh[:mm [m/d/y]]</i>]
[, USE]

OUT file

specifies a spooler or line printer for the ASAP output. If you do not specify an output file, the output is displayed on your terminal.

specifies that information for all systems is to be provided.

node

specifies the individual system for which availability information is to be displayed. If you want information for all the systems, specify *LH **. If you do not specify a node or the wild-card character (*), the default is used. The default is the system where the ASAP program was started, if not superseded by a system specified with the ASAP SYSTEM command.

line

specifies the individual Expand line of a system for which availability information is displayed. This information includes statistics about the execution history of the line, such as request counts and numbers of errors and operational state information such as Up, Down, or Connecting states. If you want all of the lines, specify *LH *.** or just *LH **. If you do not specify a line or the wild-card character (*), ASAP uses the default, which is all Expand lines for the current system.

specifies that availability information for all lines is to be displayed. If you do not specify a line or the wild-card character (*), ASAP uses the default, which is all Expand lines for the current system.

INFO

displays line attribute information. The information includes the logical device number (LDEV), the primary and backup processor identifiers, the line subtype, the NEXTSYS and time factor values, and any ASAP line collection errors.

FRAMES

displays Level 2 information/supervisory (I/S) frames sent and received and I/S frame ratios. Level 2, the link layer, is the communications interface to the controller defined by the line type.

I

represents the information frames.

S

represents the supervisory frames.

U

represents the unnumbered (nonsequenced) frames.

For Expand-over-IP and Expand-over-ATM Level 2 i/s frames have these meanings:

- IFrm(S/R) displays number of Data Frames sent and received.
- SFrm(S/R) displays number of Connect Commands sent and rcvd.
- UFrm(S/R) displays number of Connect Responses sent and rcvd.

For Expand over SERVERNET Level 2 i/s frames have these meanings:

- IFrm(S/R) displays number of Data Frames sent and received.
- SFrm(S/R) displays number of Active Connect Cmnds sent & rcvd.
- UFrm(S/R) displays number of Passive Connects sent & rcvd.

HISTO

displays a histogram of the message size distribution. The histogram provides information about message sizes less than the values of 64, 128, 256, 512, 1024, 2048, and 4096.

IP

displays the Expand statistics for Expand lines configured using either the IP transport or the ATM transport. Only lines configured using IP or ATM are displayed. Statistic counters reported are:

DFrame Number of data frames sent and received.

KB Bytes	Number of data bytes sent and received
CnCmd	Number of connect commands sent and received
CnRsp	Number of connect responses sent and received
QyCmd	Number of query commands sent and received
QyRsp	Number of query responses sent and received
MML	Number of times a memory low condition was sent to the line handler process
IvFr	Number of received frames too small for frame headers
DFr	Number of dropped frames
IA	Number of frames from an unexpected node

LINE

displays only Expand line handlers for subtype 0, 2, 3, 4, and 6. You can use this option with all other LH options.

OPERATIONS

displays the number of Level 4 operations sent and received. Level 4, the transport layer, handles and buffers messages between processes. Level 4 also maintains control blocks during sessions and formats packets into frames for transmission.

The operations include:

- CONN Connect requests
- TRCE Trace requests
- NCPM Network-control-process to network-control-process messages
- LREQ Link requests
- LCOMP Link completion messages
- CAN Cancel requests
- ACK Acknowledgment requests
- NAK Negative acknowledgments
- ENQ Enquiry requests
- PCHG Pathchange messages

PATH

displays Expand statistics only for Expand Multiline Path process subtype 1. You can use this option with all other LH options.

POOL

displays pool use. The information includes:

- Current and maximum percentage of pool used
- Number of pool failures
- Current and maximum percentage of message buffer used
- Current and maximum percentage of out-of-sequence buffer used
- Number of times the out-of-sequence timeout has elapsed

RATES

displays counts normalized to number per second.

SAMPLES *count*

specifies the number of sample periods to display. A sample period is defined by the RATE option in the SET command. You can specify 0 through 255 sample periods.

STATE

shows all attributes that have an associated state. Associated states are displayed for the Expand command without any options, and the disk command with these options: IP, PATH, LINE, RATE, PERCENT(Default).

TIME *hh:mm m/d/y*

defines the ending time of the display information in hours, minutes, month, day, and year. For example, specifying -TIME 10:15|| when the rate is 5 indicates that statistics collected from 10:10 to 10:15 are displayed.

When a -?|| appears immediately after the timestamp, it indicates that the statistics are questionable and that there was a problem obtaining the statistics. Possible reasons for questionable statistics include a down line handler or a down path.

When a -<|| appears immediately after the timestamp, the statistics were received late by the Collector process. Late statistics indicate busy Expand lines.

hh

is the hour.

mm

is the minute.

m/d/y

is the month, day, and year. Valid *m/d/y* entries are *mm/dd/yy*, *yyyy/mm/dd*, or *mm/dd/yyyy*.

USE

displays statistics about current pool usage. For percentages, see the POOL option.

RAW

displays RATES data in a dense tabular format rather than the standard ASAP sparse format.

LH Statistics

The following table lists the LH statistics and options shown by the HELP NETSTATS command.

Table 6-6. LH Statistics and Options

Statistics	Options
BCC	Total number of block-check character errors in this sample.
FRAME	Number of Level 2 I/S frames sent and received, and I/S ratios.
HISTO	Message size histogram <64 <128 <256 <512 <1024 <2048 <4096.
INFO	LDEV, primary/backup PIDs, subtype, and ASAP error, if any.
IP	Displays statistics for Expand lines using the transports IP or ATM.
L4LINKS	Total number of Level 4 links (logical messages) sent and received.
L4NAKS	Total number of Level 4 NAKs sent and received.
L4PACKETS	Total number of Level 4 packets (local+passthru) sent and received.
L4PASSTHRU	Total number of Level 4 packets (passthru only) sent and received.
LINE	Displays statistics for subtypes 0, 2, 3, 4, and 6.
NOB	Total number of no-frame-buffer errors in this sample.
OPER	Number of Level 4 operations sent and received: CONN, TRCE, NCPM,LREQ, LCOMP, CAN, ACK, NAK, ENQ, and PCHG.
PATH	Displays statistics for Multiline PATH process subtype 1.

PF	Total number of buffer pool failures in this sample.
POOL	Combined buffer pool usage consisting of message buffer and out-of-sequence buffer in percentages or kilobytes.
RATES	Counts normalized to number/second.
STATE	State of the line. Operational States are Up, Down, or Cntg (Connecting).
TIME	Statistics starting at a time other than the current time.
USE	Values in kilobytes rather than pool percentages (% is default).
RAW	Shows statistics in tabular format rather than standard display.

Table 6-7. Statistic Counters

Counter	Description
ACK	Number of Acknowledgments sent or received.
BCC	Total number of BCC errors in this sample.
CAN	Number of Cancel requests sent or received.
CnCmnd	Number of IP/ATM Connect commands sent or received.
CnRsp	Number of IP/ATM Connect responses sent or received.
CONN	Number of Connect requests sent or received.
Cur%	Percentage of combine Pool pages used Msg + Oos.
DFr	Number of IP/ATM rates of frames dropped.
DFrame	Number of IP/ATM Data Frames sent or received.
ENQ	Number of Enquiry requests sent or received.
IA	Number of IP/ATM frames received from a unexpected node.
IvFr	Number of IP and ATM rates of small frames received
KB Bytes	Number of IP or ATM data bytes sent or received.
LCOMP	Number of Link completion messages sent or received.
LkS	Total number of Level 4 Links (logical messages) sent.
LkR	Total number of Level 4 Links (logical messages) received.
LREQ	Link requests sent or received.
NCPM	\$NCP to \$NCP messages sent or received.
NkS	Total number of Level 4 Naks sent.
NkR	Total number of Level 4 Naks received.
NOB	Total number of No Frame Buffer errors in this sample.
Max%	% of maximum pages used since statistics reset applies to Pool, Message buffer (Msg Cur%), and Out of Sequence buffer (oos).

Msg Cur%	Percentage of Message buffer used.
MML	Number of IP or ATM sent a memory low condition to the LH process.
NAK	Number of Negative Acknowledgements sent or received.
OosCur%	Percentage of Out of Sequence buffer used.
PCHG	Number of Pathchange messages sent or received.
PF	Total number of buffer Pool Failures in this sample.
PkSnt	Total number of Level 4 packets (local+passthru) sent.
PkRcv	Total number of Level 4 packets (local+passthru) received.
POOL	Combined buffer Pool usage = msg + oos in % or kilobytes.
PtSn	Total number of Level 4 packets (passthru only) sent.
PtRv	Total number of Level 4 packets (passthru only) received.
QyCmd	Number of IP or ATM Query commands sent or received.
QyRsp	Number of IP or ATM Query responses sent or received.
State	Expand line handler state. Up, Down, Cntg (Connecting).
Tos	Number of Out of Sequence timeouts.
TRCE	Number of Trace or Probe requests sent or received.

Examples

These examples show the LH command with the ASAP output:

```
+LH \ISLAND                                !Default command

\ISLAND      \249   Time   State PF NoB Bcc L4naks   L4packets   L4passtru   L4links
-----
$ LHCSO      3/01   9:12   Cntg
$ LHIST      3/01   9:12?  Down
$ LHPAN      3/01   9:12   Up                7
$ LHREC  s   3/01   9:12   Up
$ LHSUP      3/01   9:12?  Down
$ LHWIN      3/01   9:12   Up                16      6                2
```

```
+LH, POOL                                !Shows pool utilization.
```

```
\CENTDIV     \051   Time   Pool   cur%  max% PF   Msg cur% max%   Oos cur% max%  tos
-----
$ LHCHI      3/01   9:09   1553    8     8     8     8
$ LHCOMM     3/01   9:09   1553    8    11     8     11
$ LHCSS1  s   3/01   9:09?  1169   28    28    28    28
$ LHCSS2  s   3/01   9:09?  1169   28    28    28    28
$ LHDOME     3/01   9:09   1553    8     8     8     8
$ LHEAST     3/01   9:09   1553    8     8     8     8
$ PATH  p    3/01   9:09?  1169   28    28    28    28
```

Additional examples without output:

```
+LH \*, S3, TIME 10:00                    !show last 3 samples starting at 10:00
+LH \*, OP                                !show level 4 operation counts
```

```
+LH \*, FRAMES      !show level 2 I/S frame counts and ratios
+LH \*, HISTO       !show message size distribution histogram
+LH \*, INFO        !show line attribute information
+LH \*, RAW         !show last sample in raw format
```


LOG Command

The LOG command displays the most recent ASAP disc log event records for ASAP in the specified node.

```
LOG [ \node ] [ blocks ]
```

blocks

optionally specifies the display of additional history. It indicates the number of log event record blocks to display.

Example

```
LOG \NEWYORK 3 !display 3 blocks of ASAP event records in \NEWYORK
```

MONITOR Command

The MONITOR command controls monitored objects (domains) within ASAP. It controls the set of monitored domains and obtains information about those domains. The MONITOR command is best explained with several different syntax displays.

To add objects to the ASAP DB for ASAP to monitor:

```
MONITOR entity [ \node. ]domain [ ,ADD [ ,OFF ] ]
```

Note

ADD is not optional if "SET MONITORADD OFF" has been entered in ASAPCONF or previously in the current ASAP C! session.

To use the PATH option when you specify OSS pathnames for files and process object filenames that do not translate automatically into ASAP domain names:

```
MONITOR FILE | PROCESS [ \node[.] ]domain PATH pathname [ , ADD [ , OFF ] ]
```

Note

When the PATH option is used the user is responsible for creating a unique ASAP domain name following the standard rules for ASAP domain names, 1-64 bytes in length, no more than 5 levels separated with a backslash, and no spaces, quotes, commas, colons, or semicolons.

To commit changes from the ASAP DB to a running copy of ASAP:

```
MONITOR [ entity ] [ \node ] ,COMMIT
```

To list or remove user-defined (ASAPX) objects:

```
MONITOR APP [ \node ] ,LIST [ ,DETAIL ]  
            [ \node.domain ] ,REMOVE [ ,FORCE ]  
            [ domain ]
```

To manipulate or report on existing objects:

```
MONITOR [ entity ] [ \node ] ,ON  
            [ \node.domain ] ,OFF  
            [ domain ] ,DELETE  
                        ,INFO [ ,DETAIL ]  
                        [ ,OBEYFORM ]  
                        ,DETAIL  
                        ,OBEYFORM  
                        ,LIST [ ,DETAIL ]
```

entity

is the name of an ASAP entity as defined in EDL.

node

is the node name the command applies to.

domain

is a form of an ASAP domain name. Domain names can represent physical devices (\$Data) or abstract representations of applications (Funds, Transfer, or Input). Some command options support the asterisk wild-card character at the end of the domain name.

The type of domain name specified in the MONITOR command varies by entity:

APP - *application*
CIPCLIM - *CLIM*
CIPMONITOR - *Monitor*
CIPPROVIDER - *Provider*
COMM - *\$line*
CPU - *cpu*
DISK - *\$volume*
entity - *application*
EXPAND - *\$line*
EXPANDIP - *\$line*
FILE - [*logical*]*\$file* | */pathname*, ADD
PROCESS - [*logical*]*\$process* | */pathname*, ADD
PROCESSBUSY - *cpu*
RDF - *primary->backup*
SPOOLER - *\$supervisor*
SWAP - *cpu*
SYSTEM - *\node*
TAPE - *\$tape*
TCPPROCESS - *\$tcpprocess*
TCPPOINT - *port number*
TCPROUTE - *\$tcpprocess.#route*
TCPSUBNET - *\$tcpprocess.#subnet*
TELSERVPROCESS - *\$telservprocess*
TELSERVSERVICE - *\$telservprocess.service*
TELSERVWINDOW - *\$telservprocess.#window*
TMF - AUDITDUMP | AUDITTRAIL | TMF | TRANSACTIONS

application

is a symbolic application name, for example SALES\SERVER\SSV02.
Applications create their own ASAP domains by using the ASAP Extension API, so only the LIST and REMOVE options are applicable for application domains.

\$line

is a line name; for example \$X25 or \$NODE1.

CLIM

is the name of a CLIM device, for example C1002111 or \$ZZCIP.C1002111.

Monitor

is the name of a CIP Monitor process, for example ZCAM0 or \$ZZCIP.ZCAM0.

Provider

is the name of a CIP Provider process, for example ZTC01 or \$ZZCIP. ZTC01.

cpu

is a processor number from 0 to 15; for example 1.

\$volume

is a disk volume name; for example \$DATA.

entity

is a symbolic application entity name; for example SALES. Applications can be represented as separate ASAP entities when they are defined using the ASAP Entity Definition Language (EDL). Only LIST and REMOVE are applicable for user-defined entities.

logical\<

is a hierarchical, logical prefix that can be used to group and create aggregates for files or processes. It can contain from one through four levels using the backslash character as a level separator, depending on the entity and type of domain name specified. For more information on using logical names, see [Section 5: File and Process Monitoring](#). SALES\ *is an example of a logical prefix.*

\$file

is a filename, subvolume name or filename pattern containing wild-card characters in the subvolume or file portions of the name. For example, \$DATA.SUBVOL, \$DATA.SUBVOL.FILE, \$DATA.SUBVOL.F*, and \$DATA.SUB*.F*.

pathname

is a process object file or OSS file specified as an OSS pathname. OSS pathnames must start with a slash character. For more information on specifying OSS pathnames to ASAP, see [Section 5: File and Process Monitoring](#).

Note

Resolving the processes running from an OSS pathname can result in excessive low-level messaging in the NonStop™ OS operating system. This method should be used judiciously in extremely high-performance application environments.

\$process

is a process name, object filename or process name pattern containing wild-card characters. For example, \$ABC, \$DATA.OBJECT.FILE and \$AB*.

Note

Resolving the processes running from an object filename can result in excessive low-level messaging in the NonStop™ OS operating system. This method should be used judiciously in extremely high-performance application environments.

primary->backup

is the primary node name followed by "->" and the backup node name. No spaces are allowed in the name, and the node names cannot contain the leading backslash character. For example, NEWYORK->CHICAGO.

\$supervisor

is the process name of the SPOOLER supervisor process for the SPOOLER subsystem you want to monitor; for example, \$SPLS.

\$node

is a node name; for example, \NODE1.

\$tape

is a tape device name; for example, \$TAPE.

\$tcpprocess

is the process name of a TCP/IP v4 process or a TCP/IPv6 Monitor process. When you monitor a TCP/IP process you monitor the process (TCPPROCESS) entity and all sub-entities that represent layers of its statistics. For example when you MONITOR TCPPROCESS \$ZTCP0 you get process stats for the TCPARP, TCPICMP, TCPICMP6, TCPIGMP, TCPIP, TCPIP6, TCPLOOP, TCPMONGQ, TCPQIO, TCP RTE, TCP SOCK and TCPUDP sub-entities. However you cannot monitor any sub-entities exclusively using the monitor command. For example you cannot MONITOR TCPARP \$ZTCP0.

port

is a valid TCP/IP port number. You cannot monitor Ports that are managed by TCP/IP processes that are not monitored. For example if you MONITOR TCPPROCESS \$ZTCP0 and then MONITOR TCP PORT 23 then ASAP will only monitor connections on port 23 that are serviced by the \$ZTCP0 process.

\$tcpprocess.#route

is the TCP/IP v4 process or TCP/IP v6 Monitor process name and route name of a TCP/IP route. If you monitor specific TCP/IP processes and then let ASAP auto-configure Routes then ASAP will only configure the routes underneath the processes you have specifically monitored.

\$tcpprocess.#subnet

is the TCP/IP v4 process or TCP/IP v6 Monitor process name and subnet name of a TCP/IP subnet. If you monitor specific TCP/IP processes and then let ASAP auto-configure Subnets then ASAP will only configure the subnets underneath the processes you have specifically monitored.

\$telservprocess

is the process name of a TELSERV process.

\$telservprocess.service

is the TELSERV process name and service name of a TELSERV service. If you monitor specific TELSERV processes and then let ASAP auto-configure Services then ASAP will only configure the services underneath the processes you have specifically monitored.

\$telservprocess.#window

is the TELSERV process and window name of a TELSERV window. If you monitor specific TELSERV processes and then let ASAP auto-configure Windows then ASAP will only configure the windows underneath the processes you have specifically monitored.

When monitored files and processes are specified using wild-card names, the wild-card specifier is contained within the domain name as a separate hierarchical level, and the actual object name is appended to the end as another hierarchical level or levels.

When processes are monitored by specifying the object filename, the object filename is contained within the domain name as a separate hierarchical level, and an actual process name is appended to the end as another hierarchical level.

COMMIT

sends a commit request to all entities or a specific entity on the local or a remote node. This makes ASAP reload its set of monitored objects for the entities on the node.

ADD

adds a domain to the set of monitored objects for a specific entity on the local or a remote node, and turns monitoring on for the domain. *entity* and *domain* are required. Wildcards are not supported.

ADD is the default if *domain* is specified without a wildcard character. ADD also outputs the domain that was added (INFO).

ON

turns monitoring on in the database for the specified domains in the specified entity. When *domain* is specified, *entity* must be specified, otherwise both are optional. Domain names can use the asterisk wildcard character as the last character in the name. For a domain to be turned on, it must first be added to the database using the ADD parameter. ON also outputs each domain turned on (INFO).

OFF

turns monitoring off in the database for the specified domain in the specified entity. When *domain* is specified, *entity* must be specified, otherwise both are optional. Domain names can use the asterisk wildcard character as the last character in the name. OFF also outputs each domain turned off (INFO).

DELETE

deletes records from the database. When *domain* is specified, *entity* must be specified, otherwise both are optional. Domain names can use the asterisk wildcard character as the last character in the name. DELETE using a wildcard domain name requires the command to be entered twice for confirmation before any action is taken.

INFO

returns information from the database about the set of monitored objects. When *domain* is specified, *entity* must be specified, otherwise both are optional. Domain names can use the asterisk wildcard character as the last character in the name. If *domain* contains a wildcard character or is not specified, INFO is the default.

DETAIL

returns detailed time information from the database for the set of monitored objects. When *domain* is specified, *entity* must be specified, otherwise both are optional. Domain names can use the asterisk wildcard character as the last character in the name.

OBEYFORM

returns sets of monitored objects in OBEY file format suitable for playback to ASAP CI. When *domain* is specified, *entity* must be specified, otherwise both are optional. Domain names can use the asterisk wildcard character as the last character in the name.

LIST

retrieves a list from the active set of objects for all entities or for a specific entity. The LIST command communicates directly with running ASAP components to retrieve object information. When *domain* is specified, *entity* must be specified, otherwise both are optional. Domain names can use the asterisk wildcard character as the last character in the name.

LIST, DETAIL

retrieves a list of the active domains from a running ASAP Server component and includes entity detail information (if any) for all entities or for a specific entity. When *domain* is specified, *entity* must be specified, otherwise both are optional. Domain names can use the asterisk wildcard character as the last character in the name.

REMOVE

removes a user-defined ASAPX domain from the monitored set of domains. An ASAPX domain must be down before it can be removed. Once removed, a domain enters a Pending Removal state until the end of the current statistics interval. It is removed once the last record is written for the domain. *entity* must be specified. *domain* can use the asterisk wildcard character as the last character in the name.

REMOVE, FORCE

removes a user-defined ASAPX domain from the monitored set of domains. FORCE removes the domain regardless of its state. Once removed, a domain enters a Pending Removal state until the end of the current statistics interval. It is removed once the last record is written for the domain. *entity* must be specified. *domain* can use the asterisk wildcard character as the last character in the name.

Considerations

Since TCP/IP and Telserv entity names are long and difficult to type ASAP supports the following abbreviations for the entity names:

TCPARP	ARP
TCPICMP	ICMP
TCPICMP6	ICMP6
TPCIGMP	IGMP
TCPIP	IP
TCPIP6	IP6
TCPLOOP	LOOP
TCPMONGQ	MONGQ
TCPPORT	PORT
TCPPROCESS	TCP
TCPQIO	QIO
TCPROUTE	ROUTE
TCPRTE	RTE
TCPSOCK	SOCK
TCPSUBNET	SUBNET
TCPUDP	UDP
TELSERVPROCESS	TELSERV
TELSERVSERVICE	SERVICE
TELSERVWINDOW	WINDOW

Since CIP entity names are long and difficult to type ASAP supports the following abbreviations for the entity names:

CIPCLIM	CLIM
---------	------

CIPMONITOR	MONITOR
CIPPROVIDER	PROVIDER

Examples

- To add and turn on the disk \$DATA:
MONITOR DISK \$DATA
- To commit all disk database changes:
MONITOR DISK, COMMIT
- To turn on all file domains already in the database:
MONITOR FILE, ON
- To turn on all domains on the local node already in the database:
MONITOR, ON
- To display all domains from the database:
MONITOR
- To turn off all disk domains:
MONITOR DISK, OFF
- To turn on all domains on \REMOTE already in the database:
MONITOR \REMOTE, ON
- To add and turn on CPU 0 on \REMOTE:
MONITOR CPU \REMOTE.0
- To turn on monitoring for RDF:
MONITOR RDF CHI->NYC

- To save the disk domain configuration:
MONITOR /OUT FILE/ DISK, INFO, OBEYFORM
- To list the current ASAPX ATM domains:
MONITOR ATM, LIST
- To list the details for ASAPX FUNDS TRANSFER:
MONITOR FUNDS\TRANSFER, LIST, DETAIL
- To monitor a spooler on the system:
MONITOR SPOOLER \$SPLG
- To remove an ASAPX user domain:
MONITOR FUNDS TRANSFER\WEST\RECEIVE, REMOVE
- To commit all change on the local node:
MONITOR, COMMIT
- To monitor a process directly enter the process name:
MONITOR PROCESS \$ABC
- To monitor a process as part of a hierarchical group of processes, precede the process name with the hierarchical name. The complete name can consist of up to five levels separated by a backslash (\), and the name can be up to 64 bytes in length. The last level must be the process name.
MONITOR PROCESS SALES\ORDERS\SERVER\\$ABC
- To create an aggregate process record for hierarchically named processes specify the name to the level where you want the aggregation to occur using the # symbol. The Process SGP automatically creates the aggregate domain record using the aggregations rules defined in the MetricRule property for each attribute in the Process EDL file.
MONITOR PROCESS SALES\ORDERS\#
- To monitor a file:
MONITOR FILE \$DATA.SUB.FILE

- To monitor all files in subvolume \$DATA.SUB that start with FIL:
MONITOR FILE \$DATA.SUB.FIL*, ADD
- To monitor an OSS file:
MONITOR FILE /home/dir/filename
- To monitor a file as part of a logical group named SALES:
MONITOR FILE SALES\DATA\\$DATA.SUB.SALES
- To dynamically monitor all processes running from an object file:
MONITOR PROCESS \$DATA.OBJECT.FILE
- To monitor processes dynamically from an object file as part of a logical process group:
MONITOR PROCESS SALES\SERVER\\$DATA.OBJECT.FILE
- To dynamically monitor all processes whose names start with \$A:
MONITOR PROCESS \$A*
- To dynamically monitor all processes whose names start with \$A as a part of the STOCKS\QUOTES logical group:
MONITOR PROCESS STOCKS\QUOTES\\$A*
- To dynamically monitor processes running from the OSS shell:
MONITOR PROCESS /bin/sh
- To dynamically monitor processes running from an OSS object file as part of a logical group named SALES\SERVER:
MONITOR PROCESS SALES\SERVER/home/finance/server.exe
- To create your own domain name and use the PATH option to monitor an OSS process:
MONITOR PROCESS SALES\SERVER\SALES2 PATH
/homedir/subdir/sales2
- To create an aggregate domain for all files on a volume named \$DATA:
MONITOR FILE \$DATA\#

- To create an aggregate-only domain for all files on a volume named \$DATA3:

```
MONITOR FILE $DATA3\##
```

- To create an aggregate process domain and suppress writing of detail records to the ASAP database, specify the name to the level where you want the aggregation to occur using the ## symbol. Detail records are not written to the ASAP database, but they are available for display from ASAP memory using the PROCESS command MEMORY option. The single aggregate record written to the ASAP database can summarize 10, 100, or even 1000 or more individual processes.

```
MONITOR PROCESS SALES\ORDERS\##
```

- To monitor a TCP/IP Process entity and its sub-entities TCPICMP, TCPIP, TCPQIO, TCP RTE and TCPUDP:

```
MONITOR TCPPROCESS $ZTCP01
```

- To monitor a specific TCP/IP route:

```
MONITOR TCPROUTE $ZTCP01.#RT1
```

- To monitor TCP/IP port 23:

```
MONITOR TCPPORT 23
```

- To monitor a TELSERV process:

```
MONITOR TELSERVPROCESS $ZTN01
```

- To monitor a specific TELSERV service:

```
MONITOR TELSERVSERVICE $ZTN01.TACL
```

- To monitor a specific CLIM device:

```
MONITOR CLIM C1002111
```

Entity	MONITOR Syntax
COMM	MONITOR COMM <i>\$devicename</i>
CIPCLIM	MONITOR CLIM <i>CLIM</i>
CIPMONITOR	MONITOR CIPMONITOR <i>Monitor</i>
CIPPROVIDER	MONITOR PROVIDER <i>Provider</i>
CPU	MONITOR CPU <i>cpu-number</i>
DISK	MONITOR DISK <i>\$devicename</i>

EXPAND	MONITOR EXPAND <i>\$linename</i>
EXPANDIP	MONITOR EXPAND <i>\$linename</i>
FILE	MONITOR FILE [<i>logical\</i>] <i>\$filename</i> <i>/pathname</i> , ADD
NODE	not applicable
PROCESS	MONITOR PROCESS [<i>logical\</i>] <i>\$process</i> <i>/pathname</i> , ADD
PROCESSBUSY	MONITOR PROCESSBUSY <i>cpu-number</i>
RDF	MONITOR RDF <i>primary-node->backup-node</i> *
SPOOLER	MONITOR SPOOLER <i>spooler-supervisorname</i>
SWAP	MONITOR SWAP <i>cpu-number</i>
SYSTEM	MONITOR SYSTEM <i>\nodename</i>
TCPPROCESS	MONITOR TCPPROCESS <i>\$tcpprocess</i>
TCPPORT	MONITOR TCPPORT <i>port</i>
TCPROUTE	MONITOR TCPRROUTE <i>\$tcpprocess.#route</i>
TCPSUBNET	MONITOR TCPSUBNET <i>\$tcpprocess.#subnet</i>
TELSERV PROCESS	MONITOR TELSERVPROCESS <i>\$telservprocess</i>
TELSERV SERVICE	MONITOR TELSERVSERVICE <i>\$telservprocess.service</i>
TELSERVWINDOW	MONITOR TELSERVWINDOW <i>\$telservprocess.#window</i>
TAPE	MONITOR TAPE <i>\$devicename</i>
TMF	MONITOR TMF <i>tmf-domain</i> **

* No backslash (\) in the node names.

** *tmf-domain* = TMF, AUDITDUMP, AUDITTRAIL or TRANSACTIONS

NODE Command

The NODE command displays two end-to-end Expand system availability statistics between the specified sending node name and the destination node name. For an explanation of NODE statistics, type HELP NCPSTATS.

<pre> NODE [/OUT <i>file/</i>] [<i>\snode</i>] [[,] \ *] [, SAMPLES <i>count</i>] [\ *] [[,] \ <i>dnode</i>] [, TIME <i>hh[:mm [m/d/y]</i>] [, NONULLOUT] </pre>

OUT *file*

specifies a spooler or line printer for the ASAP output sent. If you do not specify an output file, the output is displayed on your terminal.

snode

specifies the node of the sending Expand system. To send availability information from all the systems, specify NODE *. If you do not specify a node or the wild-card character (*), the default is used. The default is the system where the ASAP program was started, if not superseded by a system specified with the ASAP SYSTEM command.

*

specifies that information for all systems is provided.

SAMPLES *count*

specifies the number of sample periods to display. A sample period is defined by the RATE option in the SET command. You can specify 0 through 255 sample periods.

dnode

specifies the node of the destination Expand system. The default is all the systems connected to the sending node (*snode*), which has the same effect as *. If you want a particular *dnode*, specify the *dnode* by system name (such as \SUPPORT) or system number (\240).

TIME *hh:mm m/d/y*

defines the ending time of the display information in hours, minutes, month, day, and year. For example, specifying -TIME 10:15|| when the rate is 5 indicates that statistics collected from 10:10 to 10:15 are displayed.

When a -?|| appears immediately after the timestamp, it indicates that the statistics are questionable and that there was a problem obtaining the statistics. Possibly a node is down.

When a -<|| appears immediately after the timestamp, the statistics were received late by the Collector process. Late statistics indicate busy Expand lines.

hh

is the hour.

mm

is the minute.

m/d/y

is the month, day, and year. Valid *m/d/y* entries are *mm/dd/yy*, *yyyy/mm/dd*, or *mm/dd/yyyy*.

NONULLOUT

displays only nodes with nonzero end-to-end statistics.

NODE Statistics

The following table lists the NODE statistics and description/options displayed with the HELP NCPSTATS command.

Table 6-8. Node Statistics and Options

Statistics	Description/Options
LRECV	The total number of Level 4 links received
LSENT	The total number of Level 4 links sent
NSYS	The neighboring system number of the current path between the two nodes
PHOPS	The number of physical hops between the two end-to-end nodes
PLDEV	The LDEV of the current active path between the nodes
PRECV	The total number of Level 4 packets received
PSENT	The total number of Level 4 packets sent
RAW	Statistics displayed in tabular format rather than the standard display
SYSNO	The system number of the destination Expand system node
TIME	Statistics starting at a time other than the current time

Examples

These examples show the report output that you can display with the NODE command:

```
+NODE      !Show last sample for all destination nodes connected to current active source node.
```

```
\CENTDIV \051  Time  Sysno PHops PLdev NSys  LSent  LRecv  PSent  PRecv
-----
\CHITOWN  3/01  9:50  153   1   45  153           8   14   16
\MERLIN   3/01  9:50  154   6   95  197
\WINE     3/01  9:50  242   2  100  245
\DOME    3/01  9:50  245   1  100  245
\ISLAND  3/01  9:50  249   3  100  245           3    7   11
```

```
+NODE \CENTDIV, \CHITOWN, S4
```


\CENTDIV	\051	Time	Sysno	PHops	PLdev	NSys	LSent	LRecv	PSent	PRecv
\CHITOWN	3/01	9:54	153	1	45	153		6	9	11
\CHITOWN		9:53	153	1	45	153		2	7	6
\CHITOWN		9:52	153	1	45	153		6	10	11
\CHITOWN		9:51	153	1	45	153		5	8	10

OBEY Command

The OBEY command instructs ASAP to read commands from the specified file. After reaching the end of the primary OBEY file, ASAP reads commands from the original input device (or file). For more information, see the Guardian User's Guide.

```
OBEY filename
```

filename

is the name of the file containing ASAP commands.

Example

The OBEY file can contain this information to indicate which systems to start:

```
STARTUP
SET DB $DATA.ASAPDB.DB
SET RATE 10
SET EXPAND ON
SET APP ON
STARTUP \CHICAGO
STARTUP \DALLAS
STARTUP \NEWYORK
STARTUP \LA
STARTUP COLLECT
```

PAUSE Command

The PAUSE command temporarily suspends the ASAP Conversational Interface. It causes the program to suspend execution until it receives a stop, abend, or break message.

```
PAUSE
```

Example

This example shows the PAUSE command:

```
+ PAUSE
```

PROCESS Command

The PROCESS command displays availability statistics about the processes specified in the command.

```
PROCESS  [\* ] [ [.] * ] [ , SAMPLES count ]
          [\node] [ [.] $process ] [ , DETAIL ]
          [ , STATES ]
          [ , CPU n ]
          [ , TIME hh[:mm[m/d/y]] ]
          [ , AGGREGATE ] [ ,
AGGONLY | AO ]
          [ , MINSTATE [AUTO] ]
          [ [state] ]
          [ , COUNT [count] ]
          [ , MEMORY ]
          [ , EXACT ]
          [ , NOAGG ]
```

Note

The PROCESS command in ASAP Release 1 has become PROCESSBUSY in ASAP Release 2. The Selected Process function that PROCESS now performs is new in ASAP Release 2.

node

specifies the individual system for which availability statistics are displayed.

\$process

specifies the process for which availability information is displayed. Partial process names are accepted. For example, \$TX returns data from the local node for all monitored processes that have names beginning with \$TX.

SAMPLES *count*

specifies the number of sample periods to be displayed, starting with the most recent sample. A sample period is defined by the RATE option in the SET command. You can specify 0 through 255 sample periods. The default sample period number is 1.

DETAIL

specifies to return more detailed information about the specified monitored processes.

STATES

specifies to return associated OEM states for each attribute defined with STATEPAIR true in the EDL.

CPU *n*

specifies to return information for all processes on only the specified CPU.

TIME *hh:mm m/d/y*

specifies the ending time of the display information in hours, minutes, month, day, and year. For example, specifying -TIME 10:15|| when RATE is 5 indicates that statistics collected from 10:10 to 10:15 are displayed.

hh

is the hour.

mm

is the minute.

m/d/y

is the month, day, and year. Valid *m/d/y* entries are *mm/dd/yy*, *yyyy/mm/dd*, or *mm/dd/yyyy*.

AGGREGATE

includes aggregate domains in the command output. AGGREGATE is on by default.

AGGONLY

displays only aggregate domains. It can also be specified as AO or AGGREGATEONLY.

MINSTATE

displays records based on attribute states:

state

shows records with at least one attribute state equal to or higher than *state*. Valid states are 1 through 8.

AUTO

shows records with at least one attribute state greater than state 2 (OK).

COUNT

defines the maximum number of records to display.

MEMORY

retrieves data directly from ASAP memory instead of the database. ASAP uses the MEMORY option for the EDL DETAIL property of the Process Entity definition. This lets a user enter -Show Object Detailsll from ASAP Client, when aggregate only domains are used, to retrieve detail records that make up the aggregate set.

EXACT

shows statistics for only the specified domain.

Note

SAMPLES, TIME, and * cannot be used with the MEMORY option.

NOAGG

suppresses display of aggregate domains.

Examples

To show the latest statistics for the local node:

+ PROCESS

\CENTDIV Domain\Name	Status	Date	Time	Error	Cpu	Pin	Pri	Busy
\$Zoom	Up	8/13	8:33	0	0	69	160	0.12
\$Zoot	Up	8/13	8:33	0	0	252	160	0.56
Asap\Asap6\Sgp\Zoo0	Up	8/13	8:33	0	0	125	160	0.25
Asap\Asap6\Sgp\Zoo1	Up	8/13	8:33	0	1	62	160	0.13
Asap\Asap6\Sgp\Zoo2	Down	8/13	8:33?	14	0	0	0	0.00
Asap\Asap6\Sgp\Zoo3	Down	8/13	8:33?	14	0	0	0	0.00
Asap\Asap6\Sgp\Zoot	Up	8/13	8:33	0	0	252	160	0.57

To display the latest processes statistics from all nodes:

+ P *

To show four samples from node \NEWYORK ending at 10:15am:

+ P \NEWYORK, S4, TIME 10:15

To show detail statistics for process \$PROC1:

```
+ P $PROC1, DETAIL
```

To show detail statistics and their associated ASAP states for process \$SGPM:

```
+ P $SGPM, DETAIL, STATE
```

To show aggregate process statistics along with detail statistics:

```
+ P, AGGREGATE
```

To show only aggregate process statistics:

```
+ P, AGGREGATEONLY
```

To retrieve all records for all processes that are issuing alerts:

```
+ P, AGG, MINSTATE
```

To retrieve detail information about aggregate only Process domains (those ending in ##), enter this EDL DETAIL command:

```
EDL DETAIL "P ^,DE,ST,MEM,MIN,EX,CO 20"
```

PROCESSBUSY Command

Note

The PROCESSBUSY command performs the functions that PROCESS performed in ASAP Release 1. In Release 2, PROCESS performs a different function.

The PROCESSBUSY command displays process availability statistics for the specified configured nodes and CPU numbers within the network you are monitoring. The PROCESSBUSY command displays the busiest processes in the selected CPU or node in the order of busiest to least busy process so you can quickly determine which processes are consuming the most CPU cycles.

To monitor the busiest processes, use the SET PROCESSBUSY ENTRIES command to specify the number of processes you want to monitor.

To display the busiest processes, use the PROCESSBUSY ENTRIES option to specify the number of busiest processes you want to display.

All the availability information collected for each node are stored in the ASAP database so that you can obtain both current and historic process reports.

```
PROCESSBUSY (or)
PB [/OUT file/] [\*      ] [ [.] * ] [, SAMPLES count      ]
                  [\node ] [ [.] cpu ] [, ENTRIES count     ]
                                      [, STATS                ]
                                      [, STATE                 ]
                                      [, INFO                  ]
                                      [, TIME hh[:mm[m/d/y]]   ]
```

Note

The only abbreviation for PROCESSBUSY is PB. P, PR, PRO, and so on are interpreted as PROCESS, not PROCESSBUSY.

OUT *file*

specifies the spooler or line printer where the ASAP output is sent. If you do not specify an output file, the output is displayed on your terminal.

*

specifies that all process information for all systems is displayed.

node

specifies the individual system for which availability information is displayed. If you want information for all the systems, specify PROCESSBUSY *. If you do not specify a node or the wild-card character (*), the default is used. The default is the system where the ASAP program was started, if not superseded by a system specified with the ASAP SYSTEM command.

cpu

specifies the individual CPU number where availability information is displayed. If you want all the CPUs, specify CPU *. or just CPU *. If you do not specify a CPU or the wild-card character (*), ASAP uses the default, which is all CPUs in the current system.

*

specifies that availability information for all CPUs is to be displayed. If you do not specify a CPU or the wild-card character (*), ASAP uses the default, which is all CPUs in the current system.

SAMPLES *count*

specifies the number of sample periods to be displayed. A sample period is defined by the RATE option in the SET command. You can specify 0 through 255 sample periods. The default sample period number is 1.

ENTRIES *count*

specifies the number of busy processes to display in each CPU during a sample interval. You can specify 1 through 20 entries. The default number of entries is 5.

STATS

displays these detailed processbusy availability attributes:

- Time of the statistics information in database local civil time
- Percentage of CPU process time during a sample interval
- Process identification number (PIN)
- Program file name of this process
- Queue length of this process during a sample interval
- Number of messages received per second
- Number of messages sent per second
- Number of physical memory pages currently in use
- Average number of link control blocks in use or reserved by this process

STATE

shows all attributes that have an associated state. Associated states are displayed for the ProcessBusy command without any options, and the ProcessBusy command works only with no other options.

INFO

displays these detailed processbusy availability attributes:

- Time of the availability information in database local civil time
- Percentage of CPU process time during a sample interval
- Process identification number (PIN)
- Total process time in microseconds
- Total time since the process was created
- Process flags
- Wait state

- Home terminal name of the process

TIME *hh:mm m/d/y*

specifies the ending time of the display information in hours, minutes, month, day, and year. For example, specifying -TIME 10:15|| when RATE is 5 indicates that statistics collected from 10:10 to 10:15 are displayed.

hh

is the hour.

mm

is the minute.

m/d/y

is the month, day, and year. Valid *m/d/y* entries are *mm/dd/yy*, *yyyy/mm/dd*, or *mm/dd/yyyy*.

PROCESSBUSY Statistics

The following table lists the PROCESSBUSY availability information and options displayed with the HELP PCBSTATS command.

Table 6-9. Processbusy Statistics and Options

Statistics	Description/Options
Affinity	The IPU affinity of the process
Busy	The percent CPU busy on non-blade system. Percent IPU busy on blade system
CBusy	The percent CPU busy on non-blade system. Percent CPU busy averaged across IPU's in CPU on blade system (Cbusy=Busy/IPUs).
CPU Time	The process time in hours, minutes, and seconds
Elapsed Time	The total time in days since the process was created
Home Terminal	The home terminal name of the process
IPU	The IPU that the Process was last running in
LCB	The average number of link control blocks (LCBs) in use or reserved by this process
Name	The process name
Pages	The number of physical memory pages currently in use
PFR	The process flag: P=privileged, F=page fault, R=process ready to run

Pin	The process identification number
Pri	The current execution priority of this process
Program File	The program file name of this process
RcvQ	The average queue length of this process during a sample interval
Rmsg	The number of messages received per second
Smsg	The number of messages sent per second
Time	The time of the statistics information in database local civil time
Userid	The group ID and user ID of the person accessing the process
%WT	The wait state. The bits in the wait field are numbered from left to right; thus a wait state of %003 means that bits 14 and 15 are set: Wait-field .8 wait on PON CPU power on .9 wait on IOPON I/O power on .10 wait on INTR Interrupt .11 wait on LINSP INSPECT event .12 wait on LKAN Message system, cancel .13 wait on LDONE Message system, done .14 wait on LTMF TMF request .15 wait on LREQ Message system, request

Examples

- To display the last four samples for system \CHI:

```
PROCESSBUSY \CHI, S4
```

- To display four samples starting at 10:15:

```
PROCESSBUSY \*, S4, TIME 10:15
```

- To display five processes:

```
PROCESSBUSY \*, ENTRIES 5
```

- To display detailed process statistics:

```
PROCESSBUSY \*, STATS
```

- To display the detailed busy process statistics:

```
PROCESSBUSY \*, INFO
```

- These examples show the report output you can display with the PROCESSBUSY command:

```
+PB                                !Default

\ASAP    TIME    Busy Pin  Program Object Filename      Name    Pri Userid
-----%-----
00  3/01 10:51    1.6  312  $MM.ASAPXTST.DRV7      $JX17   80 255,34
      1.6  313  $MM.ASAPXTST.DRV5      $JX15   80 255,34
      1.6  314  $MM.ASAPXTST.DRV4      $JX13   80 255,34
      .4  283  $SYSTEM.SYS03.OMLAM
      .3    6  $NCP
01  3/01 10:51    1.6  312  $MM.ASAPXTST.DRV7      $JX16   80 255,34
      1.6  313  $MM.ASAPXTST.DRV5      $JX14   80 255,34
      1.6  314  $MM.ASAPXTST.DRV4      $JX12   80 255,34
      .6   13  $BR
      .3  108  $SYSTEM.SYSTEM.ASAPMON  $ZOO1  160 255,255

+
+PB \ASAP, INFO, ENTRIES #

\ASAP    TIME    Busy Pin  CPU Time    Elapsed Time PFR %WT Home Terminal
-----%-----
01  3/01 10:56    3.4   10  11:47:22    34:20:35:43 P   317 $TRM0.#A
      2.2    1   7:17:24    34:20:35:43 P   040 $TRM0.#A
      2.1    0   7:00:47    34:20:35:43 P R  000 $TRM0.#A

+
```

Note

CPU 0 was down, so no output is displayed.

RANK Command

The RANK command controls the objectives used to produce alerts and spawn actions within ASAP. It controls the set of objectives and gets information about those objectives.

```
RANK [/OUT file/] [entity | *] [\node], COMMIT
      [.| ]][domain]], simplex-goal [event] [action]
      , Boolean-goal [event] [action]
      ,LIKE domain
      ,ACTIVATE
      ,DEACTIVATE
      ,DELETE [,SIMPLEX | GOALNAMES ]
      ,INFO [,OBEYFORM ]
      ,OBEYFORM
      ,LIST

simplex-goal attr op [value] [, simplex-goal...]
```

```

Boolean-goal gname=(attr op value [ AND|OR attr op value ... ] )
                [ AND|OR Boolean-goal ] )
                gname , DELETE

event          INFO|CRITICAL [REPEAT] [NODISPLAY] [down] [-up]
                NOEMS

action        ACTION "string" | rule [WAIT wait] [NOREPEAT]

```

Use the RANK command to set or view preferred-property-threshold settings (objectives). When a ranked property threshold is exceeded, the state of the ranked property is assigned an alert state. States in ASAP describe the status of a property such as up, down, warning, or critical. ASAP displays ranked alert states in the ASAP Client, the EMS subsystem, or both.

To enable ranked property alerts in ASAP Client, set State Determination to UseStateGraphState for each ranked property in the ASAP Client Attribute Property sheet. For more information, see "Configuring ASAP" in the ASAP Client help.

To enable ranked property events in EMS subsystem, issue a SET OBJECTIVESEVENT ON command from the ASAP command interpreter before starting ASAP Server components.

The RANK command controls objectives used to produce alerts and to take actions. The MONITOR command controls which objects are monitored within ASAP.

The RANK command is also used to display objective values and actions.

The GOAL command is a synonym for the RANK command; the two are interchangeable.

entity

is the name of an ASAP entity as defined in EDL.

node

is the node name the command applies to.

domain

is any form of an ASAP domain name. Domain names can represent physical devices (\$Data) or abstract representations of applications (Funds\Transfer\Input). Some command options support the asterisk wild-card character at the end of the domain name.

COMMIT

sends a commit request to all entities or a specific entity on the local or a remote node. This request causes ASAP to reload its set of objectives for the entity or entities on the node.

simplex-goal attr op [value] [, simplex-goal ...]

*Boolean-goal gname=(attr op value [AND|OR attr op value...])
[AND|OR Boolean-goal])
gname, DELETE*

()

Parentheses are required for Boolean goals. They are used to group one or more attribute-operator-value sets. Parentheses can be nested as needed. Boolean goals are processed left to right from the innermost parentheses outward.

attr

is the name of an attribute name as defined in ASAP EDL for the entity. The attribute must be defined with StatePair Yes, and it cannot be the StatelsOp (STATUS) attribute. Note to specify an event or action on the StatelsOp attribute, specify STATUS or STATUS NOEMS instead of the <attr> attribute name.

op

is one of the following objective operators:

- > The attribute value must be greater than the objective value.
- >> The attribute value must be greater than the objective value. Used when defining more than one greater-than goal (a de-escalating goal.) Be sure to set ObjectivesEventState to HIGH or lower when using this operator.
- >>> The attribute value must be greater than the objective value. Used when defining more than one greater-than goal (a de-escalating goal.) Be sure to set ObjectivesEventState to HIGH or lower when using this operator.
- < The attribute value must be less than the objective value.
- << The attribute value must be less than the objective value. Used when defining more than one less-than goal (an escalating goal.) Be sure to set ObjectivesEventState to HIGH or lower when using this operator.

- <<< The attribute value must be less than the objective value. Used when defining more than one less-than goal (an escalating goal.) Be sure to set ObjectivesEventState to HIGH or lower when using this operator.
- = The attribute value must equal the objective value.
- <> The attribute value cannot equal the objective value.
- >= The attribute value must be greater than or equal to the objective value.
- <= The attribute value must be less than or equal to the objective value.

op has no meaning for the STATUS attribute.

value

is the stated goal for this attribute. If no value is specified, the existing value for attribute-operator is deleted. The value must match the TypeData setting as defined for the attribute in EDL. (See SHOW command.) For example, you cannot set a decimal value objective for an attribute defined as an integer.

value has no meaning for the STATUS attribute.

gname

is the name of a Boolean goal. The goal name can be from 1-32 bytes long. Boolean goals can be defined at the <entity> and the <domain> level. At a minimum, <entity> must be specified when defining a Boolean goal. The goal name must be followed by an equal sign and the Boolean goal itself must be enclosed in parentheses. A goal name specified at the <domain> level overrides a goal with the same name specified at the <entity> level.

= (*equal sign*)

The equal sign assignment is required for defining Boolean goals. It must immediately follow the goal name in a Boolean goal specification. The equal sign cannot be used for other goal specifications.

AND

is a logical operator used to combine attr-op-value sets into Boolean goals. When AND is specified it states that both sets must be true, failure of either attr-op-value comparison will cause the AND condition to fail.

OR

Is a logical operator used to combine attr-op-value sets into Boolean goals. When OR is specified it states that either set can be true for the goal to be satisfied, failure of both attr-op-value comparisons is required to cause the OR condition to fail.

event

INFO | CRITICAL [REPEAT] [NODISPLAY] [*down*] [*-up*]

INFO

configures ASAP to generate an informational event when the objective is not met.

CRITICAL

configures ASAP to generate a critical event when the objective is not met.

REPEAT

causes ASAP to repeat the event for each interval where the objective is not met. REPEAT is always on for Boolean goals.

NODISPLAY

overrides any global settings to suppress display of the event on the operator console.

down

is the event number to use for the down event, when the objective comparison fails. Events must be numbered in the range 6000 through 9999. Event number 4000 is used when this is not specified. The ASAP EMS event template for the down event is not applied when custom down events are defined.

-up

is the event number to use for the up event, which is triggered when the objective comparison returns to normal after a failure. The number specified must be in the range 6000 through 9999. Event number 4001 is used when this is not specified. The minus sign must precede the up event number even if the down event number is not specified. The ASAP EMS event template for the up event is not applied when custom up events are defined.

For escalating and de-escalating goals the up event is written when the alert level returns to normal.

Up event specifications are not supported for Boolean goals.

STATUS

When events are turned on (using SET OBJECTIVESEVENT ON), status changes in ASAP system entities default to generate one critical EMS event. The STATUS attribute specifies changes to the default behavior for system entities. Use STATUS with ASAP user-defined entities to define all event behavior for status changes.

NOEMS

Used only with the STATUS attribute. NOEMS turns off automatic event generation for status changes in ASAP system entities.

action

`ACTION string | rule [WAIT wait] [NOREPEAT]`

string

is an action string where the first word in the string is a command interpreter name such as TACL, SCF, or FUP, and the remainder of the string represents the command to be executed by that command interpreter.

The string must be enclosed in double quotation marks, and ASAP converts any single quotes found within a string to double quotes. For example, `ACTION "FUP SECURE <#OBJECT>, '<#GOAL>' "` becomes `FUP SECURE <#OBJECT>, "<#GOAL>"` when processed by the ASAP Action server.

Whenever these tokens are contained in a string, the tokens in the string are replaced with values associated with the objective, and the resulting command is executed:

<#ENTITY> The entity name

<#OBJECT>	The NonStop™ OS or OSS object name
<#DOMAIN>	The domain name
<#D1>	Leftmost part of a multilevel domain name
<#D2>	2nd part of a multilevel domain name
<#D3>	3rd part of a multilevel domain name
<#D4>	4th part of a multilevel domain name
<#D5>	5th part of a multilevel domain name
<#ATTR>	The attribute name
<#OP>	The binomial objective operator. Operators are defined using the following 2-byte values:
	GT greater than
	G2 level 2 greater than
	G3 level 3 greater than
	LT less than
	L2 level 2 less than
	L3 level 3 less than
	EQ equal
	NE not equal
	GE greater than or equal to
	LE less than or equal to
<#VALUE>	The current value of the attribute
<#STATE>	The current state of the attribute
<#LASTSTATE>	The last state of the attribute
<#GOAL>	The objective value
<#GNAME>	The name of the Boolean goal
<#GOALVAL>	Attribute, operator, goal and value in a string with no spaces where goal and value are separated by the character. For example, "PRI=150 180". "PRI = 150" is the goal setting and 180 is the current value of the PRI attribute.
<#attrGOAL>	A specific goal token for the attribute. For example if the attribute name is BUSY the name of this token will be <#BUSYGOAL> and it will contain the goal for the BUSY attribute.
<#attrOP>	A specific operator token for the attribute. For example if the attribute name is BUSY the name of this token will be <#BUSYOP> and it will contain the operator used with the BUSY

attribute.

<#attrVALUE>

A specific value token for the attribute. For example if the attribute name is BUSY the name of this token will be <#BUSYVALUE> and it will contain the value of the BUSY attribute.

When using Boolean goals where one or more attribute-operator-value settings are defined, the values returned for the <#ATTR>, <#OP>, <#VALUE>, <#GOAL> and <#GOALVAL> are returned in a list separated by vertical bars. For example consider this GOAL statement:

```
GOAL CPU CPULOAD = (BUSY < 80 AND QUEUE < 2)
```

If CPU BUSY is 90 and the QUEUE length is 3, these values will be returned for the specified tokens:

```
<#ATTR>      BUSY|QUEUE
<#OP>        LT|LT
<#VALUE>     90|3
<#GOAL>      80|2
<#GOALVAL>   BUSY<80|90|QUEUE<2|3
```

The order of the items is dependent on the evaluation order of the goal statement, which is innermost from left to right.

If an attribute is used more than once in a Boolean goal statement then the names for attribute-specific tokens like <#BUSYGOAL> will use numbers immediately following the attribute portion of the name to uniquely identify the additional tokens. For example the second <#BUSYGOAL> token will be named <#BUSY2GOAL>.

rule

is an action rule defined using the ACTION command. Action rules are stored action strings that can be applied to multiple domains.

WAIT *wait*

indicates the number of minutes ASAP should wait before retrying an action string against a particular entity and domain. The wait time also depends on the ASAP rate setting because ASAP validates objectives only once per cycle.

NOREPEAT

indicates that ASAP should not continue to repeat the action until the condition is corrected.

LIKE

sets attributes in the new or modified domain that are currently active in the LIKE domain. Any goal specified in the command then overrides corresponding LIKE domain values.

ACTIVATE

turns ranking on for the specified domain(s) in the specified entity. *entity* must be specified when domain is specified; otherwise both are optional. Domain names can use the asterisk wild-card character as the last character in the name.

Note

A double asterisk is required to select multiple File and Process records.

DEACTIVATE

turns ranking off for the specified domain(s) in the specified entity. *entity* must be specified when domain is specified, otherwise both are optional. Domain names can use the asterisk wildcard character as the last character in the name.

DELETE

deletes records from the database. *entity* must be specified when domain is specified; otherwise both are optional. Domain names can use the asterisk wild-card character as the last character in the name. DELETE using a wild-card domain name requires the command to be entered twice for confirmation before any action is taken.

SIMPLEX

deletes the simplex goals for an entity or domain, leaving any Boolean goals defined for the entity or domain.

GOALNAMES

deletes the Boolean goals for an entity or domain, leaving any simplex goals defined for the entity or domain.

INFO

returns information about objectives from the database. *entity* must be specified when domain is specified; otherwise both are optional. Domain names can use the asterisk wild-card character as the last character in the name.

OBEYFORM

returns objective information in OBEY file format suitable for playback to ASAP CI. *entity* must be specified when domain is specified; otherwise both are optional. Domain names can use the asterisk wild-card character as the last character in the name.

LIST

retrieves a list from the active set of objectives for all entities or for a specific entity. The LIST command communicates directly with running ASAP components to retrieve objective information. *entity* must be specified when domain is specified; otherwise both are optional. Domain names can use the asterisk wild-card character as the last character in the name.

Considerations

Since TCP/IP and Telserv entity names are long and difficult to type ASAP supports the following abbreviations for the entity names:

TCPARP	ARP
TCPICMP	ICMP
TCPICMP6	ICMP6
TPCIGMP	IGMP
TCPIP	IP
TCPIP6	IP6

TCPLOOP	LOOP
TCPMONGQ	MONGQ
TCPPORT	PORT
TCPPROCESS	TCP
TCPQIO	QIO
TCPROUTE	ROUTE
TCPRTE	RTE
TCPSOCK	SOCK
TCPSUBNET	SUBNET
TCPUDP	UDP
TELSERVPROCESS	TELSERV
TELSERVSERVICE	SERVICE
TELSERVWINDOW	WINDOW

Since CIP entity names are long and difficult to type ASAP supports the following abbreviations for the entity names:

CIPCLIM	CLIM
CIPMONITOR	MONITOR
CIPPROVIDER	PROVIDER

Examples

- To add or modify the record for the disk \$DATA:
RANK DISK \$DATA, BUSY < 65
- To specify that a file must be less than 90 percent full:
RANK FILE \$SYSTEM.FILE.FILE, FULL < 90
- To add or modify the CPU record, define Busy and Queue:
RANK CPU 0, BUSY < 90, QUEUE < 4
- To set the objectives for a domain:
RANK FUNDS TRANSFER, BUSY < 30, TRANSRATE >= 15, ERROR < 1

- To override global objectives and create critical events:

```
RANK FUNDS TRANSFER\WEST, BUSY > 20, BUSY < 40 CRITICAL
REPEAT
```
- To turn ranking off for a file:

```
RANK FILE \REMOTE.$DATA.FILE.FILE, DEACTIVATE
```
- To commit FILE on \REMOTE:

```
RANK FILE \REMOTE, COMMIT
```
- To save the disk objectives configuration:

```
RANK /OUT FILE/ DISK, INFO, OBEYFORM
```
- To list a domain for ATM:

```
RANK ATM\CHICAGO\ABC, LIST
```
- To deactivate a broken spooler device while it is repaired:

```
RANK SPOOLER $SPLG\DEV\SLP, DEACTIVATE
```
- To turn on all domains for all entities:

```
RANK, ACTIVATE
```
- To commit all objectives changes:

```
RANK, COMMIT
```
- To define an action to reload a CPU after it fails:

```
RANK CPU, STATUS ACTION "TAEL RELOAD <#DOMAIN>, PRIME" WAIT
15
```
- To define a goal and action to secure the NonStop™ OS userid file to "OOOO":

```
RANK FILE $SYSTEM.SYSTEM.USERID, RWEF = "OOOO" ACTION "FUP
SECURE <#DOMAIN>, '<#GOAL>' "
```
- To define a goal and action to compress a volume that is getting too full and to write a critical EMS event, but to only do the compression once every 2 hours:

```
RANK DISK, FULL < 90 CRITICAL ACT "DCOM <#DOMAIN>" WAIT 120
```
- To define custom up and down event numbers for CPU 0 getting too busy:

```
RANK CPU 0, BUSY < 90 CRITICAL 4444-5555
```

- To define a custom down event number for CPU 0 getting too busy:

```
RANK CPU 0, BUSY < 90 CRITICAL 4444
```

- To define a custom up event number for CPU 0 when it returns to normal:

```
RANK CPU 0, BUSY < 90 CRITICAL -5555
```

- To define a goal for busy for an OSS shell process aggregate-only domain:

```
RANK PROCESS /bin/sh\##, BUSY < 10
```

- To define a global goal and action to resecure any monitored file using the ASAP default RESECURE rule:

```
RANK FILE, RWEPP = "AOAO" ACTION RESECURE
```

- To define a goal that all foreign TCP/IP Port connections must originate from net 16:

```
RANK TCPPORT, FADDR = 16.0.0.0
```

- To define a goal that all foreign TCP/IP Port connections on Port 23 must originate from subnet 16.150:

```
RANK TCPPORT 23, FADDR = 16.150.0.0
```

- To define a goal to report the occurrence of any CIP Linux errors for any CLIM device:

```
RANK CIPCLIM, AGTLNXERRS = 0, SVRLNXERRS = 0
```

- To define an escalating goal for the BUSY attribute for all CPUs:

```
RANK CPU, BUSY < 50, BUSY << 70, BUSY <<< 90
```

- To define an escalating goal for the BUSY attribute for all CPUs using separate commands:

```
RANK CPU, BUSY < 50  
RANK CPU, BUSY << 70  
RANK CPU, BUSY <<< 90
```

- To define a normal simplex goal on the CPU BUSY attribute:

```
RANK CPU, BUSY << 80
```

Using the << or <<< operator is the same as using the < operator if only one goal is defined.

- To define a de-escalating goal on the TMF TRANSRATE attribute:

```
RANK TMF, TRANSRATE >30, TRANSRATE >> 20, TRANSRATE >>> 10
```

- To define an escalating goal on CPU BUSY that generates an information EMS event and then a critical EMS event as the problem worsens:

```
RANK CPU, BUSY < 50, BUSY << 70 INFO, BUSY <<< 90 CRITICAL
```

The -UPII event, if enabled, will be written when the CPU leaves the alert state by falling back below 50% busy.

- To define an escalating goal on CPU BUSY that generates specific event numbers and where the most critical event repeats while that level of problem exists:

```
RANK CPU, BUSY < 50 INFO 8881-8880  
RANK CPU, BUSY << 70 CRITICAL 8882  
RANK CPU, BUSY <<< 90 CRITICAL REPEAT 8883
```

If the CPU is 70% busy when it alerts then event 8882 will be written.

- To define an escalating goal on the SPOOLER FULLPCT attribute that takes a different action depending on how full the SPOOLER is:

```
RANK SPOOLER, FULLPCT < 50 ACTION "TACL CLEANSPL <#OBJECT>"  
RANK SPOOLER, FULLPCT << 70 ACTION "TACL DELETEJOB <#OBJECT>"  
RANK SPOOLER, FULLPCT <<< 90 ACTION "TACL FORCESPL <#OBJECT>"
```

- To define a simplex goal for all CPUs and override that goal with an escalating goal for CPU 0:

```
RANK CPU, BUSY < 60  
RANK CPU 0, BUSY < 70, BUSY << 80, BUSY <<< 90
```

- To define a Boolean goal named BUSYGOAL for all CPUs that checks the BUSY and QUEUE attributes and where failure of either will result in the goal failing:

```
RANK CPU, BUSYGOAL = (BUSY < 80 AND QUEUE < 3)
```

- To define a Boolean goal named BUSYGOAL for all CPUs that checks the BUSY and QUEUE attributes and where failure of both are required for the goal to fail:

```
RANK CPU, BUSYGOAL = (BUSY < 80 OR QUEUE < 3)
```

- To define a Boolean goal named BUSYGOAL for all CPUs that checks the BUSY and QUEUE attributes and writes a critical event if the goal fails:

```
RANK CPU, BUSYGOAL = (BUSY < 80 AND QUEUE < 3) CRITICAL
```

- To define a Boolean goal named BUSYGOAL for all CPUs that checks the BUSY and QUEUE attributes and takes an action if the goal fails:

```
RANK CPU, BUSYGOAL = (BUSY < 80 AND QUEUE < 3) ACTION "TACL  
RUN $SYSTEM.MACROS.BUSYCPU <#OBJECT> <#BUSYVALUE>  
<#QUEUEVALUE>"
```

- To define a Boolean goal named BUSYGOAL for all CPUs that checks the BUSY, QUEUE and DISP attributes:

```
RANK CPU, BUSYGOAL=(BUSY < 80 AND QUEUE < 3 AND DISP < 1000)
```

- To define a Boolean goal named RESOURCE for all CPUs that checks the BUSY, QUEUE and two PCB attributes:

```
RANK CPU, BUSYGOAL=(BUSY<80 AND QUEUE<3 AND (PCB>10 OR  
PCBX>10))
```

RDF Command

The RDF command displays RDF entity availability statistics for the specified nodes and domain names. For a description of RDF statistics, enter HELP RDFSTATS.

```
RDF [/OUT file/]  
  [ \*      ] [ [.] *      ] [ , SAMPLES count      ]  
  [ \node ] [ [.] domain ] [ , CPU                ]  
                                     [ , DETAIL            ]  
                                     [ , STATES            ]  
                                     [ , TIME hh[:mm [m/d/y]] ]  
                                     [ , VOLUMES           ]
```

node

specifies the individual system for which availability information is to be displayed. If you want information for all the systems, specify RDF *. If you do not specify a node or the wild-card character (*), the default is used. The default is the system

where the ASAP program was started, if not superseded by a system specified with the SYSTEM command.

SAMPLES *count*

is the number of samples to display per domain.

CPU

displays RDF domain or metric values in the specified CPU.

DETAIL

displays detailed RDF metric values.

STATES

displays RDF metric values and their objective states.

TIME *hh:mm m/d/y*

defines the ending time of the display information in hours, minutes, month, day, and year. For example, specifying TIME 10:15 when RATE is 5 indicates to display statistics collected from 10:10 to 10:15.

hh

is the hour.

mm

is the minute.

m/d/y

is the month, day, and year. Valid entries are mm/dd/yy, yyyy/mm/dd, or mm/dd/yyyy.

VOLUMES

displays the associated disk volumes.

Examples

- To show the latest availability information for the default node:
RDF
- To show RDF domains on all nodes:

RDF *

\TANDA	Domain\Name\Hierarchy	Status	Date	Time	Error
Dome->	Tanda\Imagetrail\Data4	Running	5/11	9:37	0
Dome->	Tanda\Purger\Purg	Running	5/11	9:37	0
Dome->	Tanda\Receiver\Rec1	Running	5/11	9:37	0
Dome->	Tanda\Updater\Up01	Running	5/11	9:37	0
Dome->	Tanda\Updater\Up02	Running	5/11	9:37	0
Dome->	Tanda\Updater\Up03	Running	5/11	9:37	0

\DOME	Domain\Name\Hierarchy	Status	Date	Time	Error
Dome->	Tanda\Extractor\Ext1	Running	5/11	9:38	0
Dome->	Tanda\Monitor\Mntr	Running	5/11	9:38	0

+

- To show detailed RDF metrics:

RDF *, DETAIL

\TANDA	Domain\Name\Hierarchy	Status	Date	Time	Error
Dome->	Tanda\Imagetrail\Data4	Running	5/11	9:39	0
Dome->	Tanda\Purger\Purg	Running	5/11	9:39	0
Dome->	Tanda\Receiver\Rec1	Running	5/11	9:39	0
Dome->	Tanda\Updater\Up01	Running	5/11	9:39	0
Dome->	Tanda\Updater\Up02	Running	5/11	9:39	0
Dome->	Tanda\Updater\Up03	Running	5/11	9:39	0

\DOME	Domain\Name\Hierarchy	Status	Date	Time	Error
Dome->	Tanda\Extractor\Ext1	Running	5/11	9:39	0
Dome->	Tanda\Monitor\Mntr	Running	5/11	9:39	0

continued...

RTDTime	PCpu	BCpu	Pri
0:00:00	0	0	0
0:00:00	1	0	185
0:00:00	1	0	185
0:00:00	1	0	185
0:00:00	1	0	185
0:00:00	0	1	185
0:00:00	2	3	185
0:00:00	3	2	185

+

- To show related disk volumes:

RDF *, VOLUME

\TANDA	Domain\Name\Hierarchy	Aud Vol	Img Vol	Pri Vol	Bkp Vol
Dome->	Tanda\Imagetrail\Data4	n/a	\$DATA4	n/a	n/a

```

Dome->Tanda\Purger\$\Purg          n/a      n/a      n/a      n/a
Dome->Tanda\Receiver\$\Rec1         n/a      $DATA2   n/a      n/a
Dome->Tanda\Updater\$\Up01          n/a      $DATA4   $DATA03  $DATA
Dome->Tanda\Updater\$\Up02          n/a      $DATA4   $DATA04  $DATA1
Dome->Tanda\Updater\$\Up03          n/a      $DATA4   $DATA05  $DATA4

```

```

\DOME      Domain\Name\Hierarchy  Aud Vol  Img Vol  Pri Vol  Bkp Vol
-----
Dome->Tanda\Extractor\$\Ext1       $AUDIT   n/a      n/a      n/a
Dome->Tanda\Monitor\$\Mntr         $AUDIT   n/a      n/a      n/a

```

+

To show statistics in RAW format:

RDF *, RAW, DETAIL

V2.10102 Num Domain\Name\Hierarchy

```

-----
-
\TANDA    230  Dome->Tanda\Imagetrail\$\Data4
\TANDA    230  Dome->Tanda\Purger\$\Purg
\TANDA    230  Dome->Tanda\Receiver\$\Rec1
\TANDA    230  Dome->Tanda\Updater\$\Up01
\TANDA    230  Dome->Tanda\Updater\$\Up02
\TANDA    230  Dome->Tanda\Updater\$\Up03
\DOME     245  Dome->Tanda\Extractor\$\Ext1
\DOME     245  Dome->Tanda\Monitor\$\Mntr

```

continued...

Status	Dateymd	Time	v	Et	AI	Seqnce	RByte	Addr
Running	2001/05/11	09:45	1	01	0	4		0
Running	2001/05/11	09:45	1	01	0	0		0
Running	2001/05/11	09:45	1	01	0	4		0
Running	2001/05/11	09:45	1	01	0	4	14270464	
Running	2001/05/11	09:45	1	01	0	4	14270464	
Running	2001/05/11	09:45	1	01	0	4	14270464	
Running	2001/05/11	09:45	1	01	0	136	10434552	
Running	2001/05/11	09:45	1	01	0	136		0

+

To show four samples starting at 10:15:

RDF \NEWYORK, S4, TIME 10:15

To show all statistics for a specific domain:

RDF LA->NewYork\Extractor\\$\Ex2, DETAIL, RAW

V2.10102 Num Domain\Name\Hierarchy

```

-----
-
\DOME     245  Dome->Tanda\Extractor\$\Ext1

```

continued...

```

Status          Dateymd      Time  v Et Error RTDsecs  RTDTime  PCpu BCpu
Pri
-----
Running         2001/05/11 09:50 1 01      0         0  0:00:00  2   3 185

```

continued...

```

AI Seqnce RByte Addr
--
0    136   10435888
+

```

SET Command

The SET command displays and sets these data collection and monitoring options:

SET	[ACTION	ON OFF]
	[ACTOBJECT	filename]
	[ACTSHELL	filename]
	[APP	entity-option]
	[ASAPLOG	filename]
	[BACKUPCPU	cpu *]
	[BUFFERED	ON OFF]
	[CIP	entity-option]
	[CLEANTIME	time]
	[COLLECT	\collect-node]
	[COMM entity-option	[, entity-option],]
	[CPU	BUSY % QUEUE q IO io SWAP swaps]
	[DB	db-file-prefix]
	[DBMAXDAYS	[[entity] days]]
	[DBMAXDAYSTOPURGE	days]
	[DBROLLOVERSUBVOLPREFIX	prefix]
	[DBROLLOVERFILEPREFIX	prefix]
	[DISK	entity-option]
	[EDL	filename]
	[EXPAND	entity-option]
	[FILE	entity-option]
	[HELPFILE	filename]
	[ID	\$asap-pid]
	[INSPECT	ON OFF]
	[IOCONTROL	count]
	[LH	NAK nak count NOBUF count BCC count]
	[LOGFILE	filename]
	[LOGGING	ON OFF]
	[LOGINTERVAL	minutes]
	[LOGSUPPRESS	ON OFF]
	[MONITORADD	ON OFF]
	[NODE	entity-option]
	[NODEDOWNALERT	ON OFF]
	[OBJCOLLECT	collect-object]
	[OBJECTIVESAUDIT	ON OFF]
	[OBJECTIVESDB	filename]
	[OBJECTIVESEVENT	ON OFF]

[OBJECTIVESEVENTUP	ON OFF]
[OBJECTIVESEVENTCONSOLE	ON OFF]
[OBJECTIVESEVENTCONSOLEUP	ON OFF]
[OBJECTIVESEVENTSTATE	OEM-state]
[OBJECTIVESEVENTSUBJECT	ON OFF]
[OBJECTIVESEVENTMAX	max events per domain per interval]
[OBJECTIVESRANK	STANDARD PERCENT HISTORICAL]
[OBJMONITOR	monitor-object]
[PARTITION	partition-descriptor]
[PRIMARY	cpu *]
[PRIORITY	server-priority]
[PROCESS	entity-option]
[PROCESSBUSY	ENTRIES count SUBSAMPLES count]
[PROXY	ON OFF]
[PROXYCONFIG	filename]
[PROXYCPU	number]
[PROXYOBJECT	object-filename]
[PROXYPARAM	proxy-parameter]
[RATE	minutes]
[RDF	entity-option]
[REMOTESECURITY	ON OFF]
[RETAIN	option]
[SLEEPTIME	time,time]
[SPOOLER	entity-option]
[SWAP	entity-option]
[SYNC	minutes,frequency]
[TAPE	entity-option]
[TERM	logterm ON OFF]
[TCP	entity-option]
[TMF	entity-option]
[TRACE	ON OFF]

ACTION ON | OFF

defines whether the ASAP action processing components are enabled.

ON | OFF

enables or disables ASAP action processing components. The default value is OFF.

When action is on, ASAP will execute user defined actions associated with the RANK ACTION option. Actions are taken for objects that have failed objectives and whose objective state meets or exceeds the SET ObjectivesEventState setting.

For more information about how to set goals and actions, see the [RANK command](#) (also known as the GOAL command).

SET ACTION ON indicates when a RANK or GOAL objective is not met, and that objective has an ACTION rule associated with it, the action rule will have macro substitution performed on the rule string, and the expanded action string will be executed. See HELP RANK for details on ACTION objectives.

To enable ACTION processing, set ACTION on prior to starting the ASAP monitor via the START MONITOR command. The monitor will keep a persistent action server running at all times, even in the event of a failed CPU.

The ASAP action server has a process name of \$*id*X, where *id* is the ASAP *id* defined via the SET ID option.

SET ACTION OFF means the RANK/GOAL ... ACTION *string* associated with an objective is not performed when objectives are not satisfied.

ACTOBJECT *action-server-program-filename*

defines the action server program object filename. The default is SEEVIEW.

ACTSHELL *action-script-filename*

defines the action server script filename. The default is ASAPACT.

APP *entity-option* [, *entity-option*]...

determines settings and parameters for the APP entity.

CPU *cpu-num*

defines the CPU that the APP SGP process will reside in.

ON | OFF

enables or disables network-wide App availability statistics reporting.

The default is OFF.

OBJECT ASAP *APP-program-file-name*

defines the ASAPX SGP program object file name. If ON is specified, the program is started automatically on each node by the ASAP Monitor.

The default is \$SYSTEM.SYSTEM.ASAPXSGP.

PARAMETERS "*APP- [APP-param]...*" | null

specifies the APP-specific parameter string to be passed to the ASAPXSGP process at startup. The parameter list must be enclosed in quotation marks. The default is null. The *APP-param* values are:

CONFIG *filename*

defines the APP configuration file.

TRACE

The default is \$SYSTEM.SYSTEM.ASAPXCNF.

ASPLOG *filename*

defines the name of the Monitor or Collect error logfile. This *filename* must be the same on all nodes that will have Monitor or Collect processes executing on them.

The default is \$SYSTEM.SYSTEM.ASPLOG.

BACKUPCPU *cpu* | *

defines the preferred backup CPU number for the Collect or Monitor process started by the STARTUP command. If this CPU is unavailable, the next available CPU is used. * indicates that if a CPU is available, the Monitor or Collect process assigns the next available CPU as the backup CPU.

BUFFERED ON | OFF

specifies whether the information written to the database is buffered. ON specifies buffered writes. The default is OFF.

CIP *entity-option*

defines startup options for CIP monitoring within ASAP.

ON | OFF

turns CIP monitoring on and off. The default is OFF.

OBJECT *filename*

specifies the name of the CIP SGP program file. The default is \$SYSTEM.SYSTEM.ASAPCIP.

PARAMETERS "CIP-param [CIP-param]... " | null

specifies the CIP specific parameter string to be passed to the ASAPCIP process at startup time. The default is the <null> string. The Parameter list must be enclosed in double quotes.

CIP-param options are:

TRACE

turns on \$RECEIVE trace to the EMS and ASAP log files.

RATE *minutes*

defines the interval, in minutes, that CIP statistics are reported back to the Collector. The default is the ASAP RATE. The range is from 1 to 120 minutes.

SCPNAME *\$name*

defines the SCP process name the SGP will use to retrieve statistics from the CIP subsystem. If no SCPNAME is specified, the SGP will attempt to start one but security concerns could prevent a successful start of an SCP process.

MAXCLIM *n*

defines the maximum of number of CLIM domain records that can be processed by the SGP. Domains are either autoconfigured or entered using the MONITOR command. The default is 128 domains. The maximum is 32767.

MAXMON *n*

defines the maximum of number of MONITOR domain records that can be processed by the SGP. Domains are either autoconfigured or entered using the MONITOR command. The default is 64 domains. The maximum is 32767.

MAXPROV *n*

defines the maximum of number of PROVIDER domain records that can be processed by the SGP. Domains are either autoconfigured or entered using the MONITOR command. The default is 256 domains. The maximum is 32767.

The ASAP monitor does not check the contents of the parameter string. All parameter data is passed to the ASAPCIP process without modification.

To set PARAMETERS to null, enter:

```
SET CIP PARAMETER
```

CPU *cpu-num*

defines the CPU that the CIP SGP process will reside in.

Examples

```
SET CIP OBJECT $SYSTEM.ASAP.ASAPCIP, PARAMETERS "SCPNAME  
ZNET", CPU 2
```

```
SET CIP PARAM "SCPNAME $ZNET"
```


CLEANTIME *time*

defines the time of day in hours and minutes that database self-cleaning begins. The default is 0000 (midnight). The RETAIN option determines what happens at cleantime.

COLLECT *collect-node*

defines the network node where the Collect process resides. The Monitor process reports statistics to the Collect process on the specified node. The default is the system where you issued the START COLLECT command.

COMM *entity-option* [, *entity-option*],

determines settings, parameters, alarm threshold settings for the COMM entity.

CPU *cpu-num*

defines the CPU that the COMM SGP process will reside in.

ON | OFF

ON enables network-wide comm statistics reporting. OFF disables network-wide comm statistics reporting. The default is OFF.

OBJECT ASAP *COMM-program-file-name*

defines the ASAP/COMM program object file name. The program is started automatically on each node by the ASAP Monitor if ON is specified. The default is \$SYSTEM.SYSTEM.ASAPCMM.

PARAMETERS "*COMM-param* [*COMM-param*]..." | *null*

specifies the communication-specific parameter string to be passed to the ASAPCMM process at startup time. The parameter list must be enclosed in quotation marks. The default is null. COMM-param values are:

CPU *cpu-number*

defines the CPU where the MEASFH process resides. If an unavailable CPU is selected, the ASPCMM process selects the next available CPU. The default is the same as the CPU that the ASAP CI is executing in.

NOMEAS

prevents COMM performance statistics from being collected. COMM state information, and the COMM information State, LDEV, PIN, Type, Configuration info for D-Series or G-Series, and the program object file are collected during each sample period.

RATE *minutes*

defines the interval, in minutes, that COMM statistics are reported back to the Collector. ASAPCMM uses the interval provided by the operator. The default is the ASAP RATE.

VOLUME [*\$volume*][.][*subvolume*]

specifies the location of the Measure Data file. The default is \$SYSTEM.SYSTEM.

SUBVOL *subvolume*

specifies the subvolume location of the Measure Data file. The default subvolume is SYSTEM.

SWAPVOL [*\$volume*]

specifies the location for the Measfh swap volume. The default is \$SYSTEM.

Example

```
+SET COMM ON, CPU 3, PARAMETERS "SUBVOL ASAPMEAS, RATE 15"
```

DB *db-file-prefix*

defines the database volume, subvolume, and file prefix of the collector server statistics files. Database file names are formed by appending the statistic entity name to the *db-file-prefix*. It must be a local name. The default prefix is \$SYSTEM.ASAP.DB.

The DB option resets all partition configuration information, so define your database before configuring partition information.

DBMAXDAYS [[*entity*] *days*]

Defines or displays the maximum number of *days* of statistical history retained in the ASAP data base for each entity.

entity

specifies the entity.

days

specifies the max number of <days> of statistical history that will be retained for that entity.

This option works in conjunction with the SET RETAIN ROLLOVER option allowing database record sets on a per entity basis to be automatically rolled-over and saved on a daily basis.

The SET DBMaxDays option is only meaningful if SET RETAIN ROLLOVER is enabled. Note that setting DBMaxDays to a nonzero value also sets the RETAIN option to ROLLOVER. See HELP SET RETAIN for details.

SET DBMAXDAYS *days* without an entity name asserts the maximum number of default *days* retained for entities, and enables the option when the *days* value is nonzero. For example, SET DBMAXDAYS 7 retains 1 week of data for all entities, unless further qualified.

SET DBMAXDAYS without specifying *entity* or *days* causes the DBMaxDays settings for each entity to be displayed.

SET DBMAXDAYS 0 disables the option and resets all ROLLOVER settings.

When SET DBMaxDays is greater than zero, rollover files are saved in the SET DBRolloverSubvolPrefix subvolume with rollover YEAR appended to the rollover subvol prefix; for example, \$DATA.ASAP_{YYYY}.* or \$DATA.ASAP2004. When DBMaxDays is enabled, the ASAP Collector limits the number of daily filesets for each entity up to *days*. Since rollover occurs daily, this option allows ASAP to automatically maintain a most recent number of *days* history without the need for manual intervention.

Examples

```
SET RETAIN ROLLOVER    -- enable SET DBMAXDAYS
SET DBMAXDAYS 1        -- retain 1 day of ASAP data as default
SET DBMAXDAYS Cpu 14   -- retain 14 days of ASAP CPU data
SET DBMAXDAYS Disk 7   -- retain 7 days of ASAP DISK data
SET DBMAXDAYS App 0    -- retain 0 days of ASAP APP data
SET DB $DATA.ASAPDB.DB
```

The set options above will retain 1 day of ASAP data for all entities except CPU, DISK, and APP which will have 14, 7, and 0 days of history retained respectively. For the settings given above, on January 3, 2004, the DISK data base files will be named as follows when the DB rollover occurs:

```
$DATA.ASAP2004.Z0102DSK -- 2004/01/02
$DATA.ASAP2004.Z0101DSK -- 2004/01/01
$DATA.ASAP2003.Z1231DSK -- 2003/12/31
$DATA.ASAP2003.Z1230DSK -- 2003/12/30
$DATA.ASAP2003.Z1229DSK -- 2003/12/29
$DATA.ASAP2003.Z1228DSK -- 2003/12/28
$DATA.ASAP2003.Z1227DSK -- 2003/12/27
```

DBMAXDAYSTOPURGE *days*

Defines the maximum number of old daily filesets that will be purged at each database cleanup rollover. This option limits the number of daily file sets that will be purged for a given rollover.

days

specifies the number of daily purge file attempts for each entity. A value other than 1 causes the rollover algorithm to reattempt file purges on subsequent *days*. A *days* value other than 1 allows a rolled-over file that was open for analysis to have purge attempts for that file to occur on subsequent *days*. Thus if *days* were 3, there would be 3 daily attempts to purge the file for each entity. The value of *days* should typically not exceed 7. The maximum value for *days* is 60.

Consider the following settings:

```
+ SET DBMaxDays 7
+ SET DBMaxDaysToPurge 1
+ SET DBRolloverSubvolPrefix ASAP
+ SET DBRolloverFilePrefix Z
```

Given these above settings, a database rollover that occurred on January 8, 2004 would purge the following fileset:

```
- ASAP2003.Z1231*
```

Using the same settings above, with DBMaxPurges set to 3 instead. A rollover that occurred on January 8, 2004 would attempt to purge the following filesets:

```
- ASAP2003.Z1231*
- ASAP2003.Z1230*
- ASAP2003.Z1229*
```

DBROLLOVERSUBVOLPREFIX *prefix*

prefix

This option defines the first 1 through 4 alphanumeric characters of the database rollover subvolume name. The option is only meaningful when SET RETAIN ROLLOVER is specified. For more details on SET RETAIN options, see HELP SET RETAIN.

The Default value is ASAP. Thus if SET RETAIN ROLLOVER is specified, and SET DB equals \$DATA.ASAPDB.DB, and SET DBRolloverSubvolPrefix is ASAP, rollover files will be named using the following subvol-file template:

```
$DATA.ASAPyyyy.ZmmdE
```

Where:

prefix

defines the 1 through 4 alphanumeric character prefix. For example, ASAP.

YYYY

indicates year rollover occurred. For example, 2004.

mm

indicates the month rollover occurred. For example, 12.

dd

indicates day rollover occurred. For example, 31.

EEE

indicates three letter entity acronym. For example, CPU, DSK, or EXP.

SET DBRolloverSubvolPrefix ASAP results in the following filename for a file containing CPU statistics from 2004, December 31:

```
$DATA.ASAP2004.Z1231CPU
```

SET DBROLLOVERFILEPREFIX *prefix*

Reserved for development use only.

DISK *entity-option* [, *entity-option*],

determines settings, parameters, alarm threshold settings for the DISK entity.

CPU *cpu-num*

defines the CPU that the DISK SGP process will reside in.

ON | OFF

ON enables network-wide disk statistics reporting. OFF disables network-wide disk statistics reporting.

The default is OFF.

OBJECT ASAP DISK-*program-file-name*

defines the ASAP/DISK program object file name. The program is started automatically on each node by the ASAP Monitor if ON is specified.

The default is \$SYSTEM.SYSTEM.ASAPDSK.

PARAMETERS "*DISK-param* [*DISK-param*}..." | *null*

specifies the disk-specific parameter string to be passed to the ASAPDSK process at startup time. The parameter list must be enclosed in quotation marks. The default is null. *DISK-param* values are:

CPU *cpu-number*

defines the CPU where the MEASFH process resides. If an unavailable CPU is selected, the ASPDSK process selects the next available CPU.

The default is the same as the CPU that the ASAP CI is executing in.

NOMEAS

prevents disk performance statistics from being collected. Disk state information, and the disk information GBUsed, Capacity, MB Available, and %MBUsed are collected during each sample period. On D-series nodes, disk controller and unit information is displayed. On G-series nodes, Group, Slot, and Module information is displayed.

RATE *minutes*

defines the interval, in minutes, that disk statistics are reported back to the Collector. The default is the ASAP rate.

VOLUME [*\$ volume*] [*.*] [*subvolume*]

specifies the location of the Measure data. The maximum file size is approximately 60 KB. This file is used by the Measure subsystem to obtain DP2 disk counters. The file is deleted when the ASAP Monitor stops the ASPDSK process.

The default location is \$SYSTEM.SYSTEM.

SWAPVOL [*\$ volume*]

specifies the location for the MEASFH swap volume.

The default is \$SYSTEM.

Example

```
+SET DISK ON, PARAMETERS "SUBVOL ASAPMEAS, RATE 15"
```

EDL *filename*

defines and compiles a file containing ASAP Entity Definition Language (EDL) statements. EDL is used within ASAP to define all entities and their attributes. You can enter multiple SET EDL commands, but a maximum of 10 EDL files are

supported for user-defined entities used with the ASAP Extension. The default is
SET EDL \$SYSTEM.SYSTEM.ASAPEDL.

EXPAND ON | OFF

controls network-wide Expand line-handler availability reporting.

CPU *cpu-num*

defines the CPU that the EXPAND SGP process will reside in.

ON | OFF

ON enables or disables network-wide Expand line-handler availability reporting.

The default is OFF.

OBJECT *ASAPEXP-program-file-name*

defines the ASAPEXP program object file name. If ON is specified, this program automatically starts on each node by the ASAP Monitor.

The default is \$SYSTEM.SYSTEM.ASAPEXP.

PARAMETERS " *expand-param* [*expand-param*]..." | null

specifies the NODE-specific parameter string to be passed to the ASAPEXP process at startup time. You must enclose the parameter list in quotation marks.

The default is null.

RATE *minutes*

defines the interval, in minutes, that Expand availability statistics are reported back to the Collector.

FILE *entity-option*

defines the settings and parameters for the FILE entity.

entity-option is one of:

CPU *cpu-num*

defines the CPU that the FILE SGP process will reside in.

ON | OFF

ON enables network-wide FILE status reporting. OFF disables network-wide FILE status reporting. The default value is ON.

OBJECT <ASAP/FILE-program-file-name>

defines the *ASAP/FILE program object filename*; this program is automatically started on each node by the ASAP monitor. The default is \$SYSTEM.SYSTEM.ASAPFIL.

PARAMETERS "*FILE-param* [*FILE-param*]... " | *null*

specifies the FILE specific parameter string to be passed to the ASAPFIL process at startup time. The default is the *null* string.

The Parameter list must be enclosed in double quotes.

The *FILE-param* option is:

AGG

when present, ASAP creates aggregate and/or state propagated group records for files defined as a part of logical file groups.

AGGONLY

when present, ASAP creates aggregate and/or state propagated group records for files defined as a part of logical file groups, and it writes only these group records to the ASAP database.

MINSTATE *state*

defines the state at which the File SGP should consider a domain to be in an alert state. Valid values are 2 through 8, the default is 3.

MINONLY *state*

When a MINONLY state is specified, the File SGP operates in exception mode, writing only alerting detail domain records to the database. MINONLY is used to determine the state at which a domain is considered to be alerting and ASAP writes only those domain records to the database, removing them after they return to normal. A domain is considered to be alerting if ANY attribute state is greater than or equal to MINONLY. MINONLY is not applied to aggregate domains.

NORECS

In some rare cases no file history is required, and users do not want to monitor files using ASAP Client. For these cases, a user can specify NORECS to cause the File SGP to stop writing all detail process records to the database. The SGP still monitors the files and generates EMS events (if enabled) when they alert, but it does not write records to the ASAP database for viewing by the ASAP Client or for historical tracking.

TRACE

turns \$Receive tracing on in the File SGP. The SGP generates a log message for each read or reply.

RATE #mins

sets a specific rate for the File SGP. See SET RATE.

MAXAGGS

defines the maximum number of aggregate domains. The File SGP will allocate memory to support that number of aggregate domains. Valid values are 2 through 32767. ASAP will always allocate 1 less domain than you specify. The default setting can vary by ASAP version.

MAXDOMAINS

defines the maximum number of monitored domains. The File SGP will allocate memory to support that number of monitored domains. Valid values are 2 through 32767. ASAP will always allocate 1 less domain than you specify. The default setting can vary by ASAP version.

MBYTES

defines the size in megabytes of the memory pool allocated to store domains that are dynamically resolved at each interval. The default setting can vary by ASAP version.

Note

The MONITOR FILE, LIST, DETAIL command will return information about the maximum value parameters and the memory pool utilization.

The ASAP monitor does not check the contents of the parameter string. All parameter data is passed to the ASAPFIL process without modification.

Examples

```
+SET FILE ON, OBJECT $SYSTEM.ASAP.ASAPFIL, PARAM "RATE 10"
```

To set PARAMETERS to null, enter either of:

SET FILE PARAM

SET FILE PARAM, *entity-option*

HELPPFILE *filename*

defines the name of the ASAP help file. This name defaults to ASAPHELP in the same subvolume as the ASAP object file. The structure of the ASAP help file is:

Keywords start in column 3, with the help text for the keyword following in subsequent lines that have text in column 4 or greater. Help text continues until another line with text in column 3 is encountered.

All lines starting with a TGAL backslash are ignored.

ID *\$asap-pid*

defines the ASAP process ID name set. This name set identifies a collection of ASAP processes that are operating together to gather ASAP performance statistics and operational state information. The default is \$ZOO. Using the ID lets you operate several ASAP programs in the same node or resolve a name conflict. This ID is used to form process names:

Logical Function	Process Name	Default
Collector	<i>\$ASAP-pid S</i>	\$ZOOS
Monitor (node)	<i>\$ASAP-pid M</i>	\$ZOOM
SGP (CPU)	<i>\$ASAP-pid 0-F</i>	\$ZOO 0 through F
APP SGP	<i>\$ASAP-pid H</i>	\$ZOOH
DISK SGP	<i>\$ASAP-pid J</i>	\$ZOOJ
NODE SGP	<i>\$ ASAP-pid G</i>	\$ZOOG

INSPECT ON | OFF

provides a debugging tool for use of the ASAP developers only. Use this option only when assisting HPE in problem diagnosis.

ON

enables debugging.

OFF

disables debugging. The default is OFF.

IOCONTROL *count*

defines the maximum number of output lines that can be written to the output file for any command before a prompt is issued.

This option is used to perform flow control between the ASAP Client and Server. By limiting the maximum number of output lines per I/O, client/server loading on communications infrastructure is controlled.

The More? prompt is a modeless prompt. Anything other than a carriage return is interpreted as a command. To continue displaying statistical data for the current command, press Return with no spaces. If Node statistics are being collected, enter NO at the prompt to display Node statistics.

The default is 72 lines of output. To disable this feature, enter SET IOCONTROL 0.

LOGFILE *filename*

defines the name of the file used for ASAP CI I/O capture. LOGGING ON/OFF controls access to the file.

The default is the current `$volume.subvolume.logfile`.

LOGGING ON | OFF

determines settings for the capture of ASAP CI I/O.

ON

enables the capture of ASAP CI I/O to a file defined by LOGFILE *filename*.

OFF

disables the capture of ASAP CI I/O.

The default is OFF.

LOGINTERVAL *minutes*

defines the interval, in minutes that duplicate ASAP error events are suppressed. During an interval, only the first occurrence for a given error event is displayed. The display of any subsequent error events during the interval are completely suppressed, both in the ASAPLOG, and in the EMS subsystem. ASAP INFO messages are not suppressed. The interval range is 0 through 1440. 0 completely suppresses ASAP error events. When LOGSUPPRESS is on, a summary record for each error type is written in the ASAPLOG and in the EMS subsystem indicating how many times the error occurred during the suppression period. The default is 60.

Examples

```
SET LOGINTERVAL 60      Suppress duplicate error events for 60
minutes
SET LOGINTERVAL 1440   Log each error type only once per day
```

```
LOGSUPPRESS ON | OFF
```

controls the suppression of duplicate ASAPLOG file error events and EMS tokenized error events. The default is OFF.

```
OFF
```

disables duplicate ASAP error event suppression.

```
ON
```

enables duplicate ASAP error suppression. The LOGINTERVAL setting determines the time duration for duplicate event suppression.

```
MONITORADD ON | OFF
```

```
ON
```

sets the default for MONITOR *entity domain* to ADD the domain to the set of monitored objects. This is the default.

```
OFF
```

sets the default for MONITOR *entity domain* to be INFO, to list the domain if it belongs to the set of monitored objects in the database.

Examples

By default, this command adds \$DATA as a disk to be monitored. If \$DATA is the first disk to be entered, the command also turns off auto-configuration of all disks.

```
MONITOR DISK $DATA
```

To turn off the default MONITOR behavior:

```
SET MONITORADD OFF
```

This command returns information about \$DATA if it is found in the database, but it does not automatically add it to the set of monitored disk objects:

```
MONITOR DISK $DATA
```

This command adds \$DATA as a disk to be monitored regardless of the SET MONITORADD setting:

```
MONITOR DISK $DATA, ADD
```

```
NODE entity-option [, entity-option]...
```

determines settings and parameters for the NODE entity.

```
CPU cpu-num
```

defines the CPU that the NODE SGP process will reside in.

```
ON | OFF
```

enables or disables network-wide node availability reporting.

The default is OFF.

```
OBJECT ASAP/NCP-program-file-name
```

defines the ASAP/NCP program object file name. If ON is specified, this program automatically starts on each node by the ASAP Monitor.

The default is \$SYSTEM.SYSTEM.ASAPNCP.

```
PARAMETERS " node-param [node-param]..." | null
```

specifies the NODE-specific parameter string to be passed to the ASAPNCP process at startup time. The parameter list must be enclosed in quotation marks. The default is null.

```
RATE minutes
```

defines the interval, in minutes, that NCP statistics are reported back to the Collector.

The default is 10 minutes.

Examples

These two examples show the SET NODE command:

```
+ SET NODE ON, OBJECT $SYSTEM.ASAP.ASAPNCP
```

```
+ SET NODE ON, OBJECT $DATA.TEST.ASAPNCP, PARAM "RATE 10"
```

```
NODEDOWNALERT ON | OFF
```

controls whether the ASAP Collector will generate a node down/up alert when a node monitored by ASAP goes down/up.

ON | OFF

indicates whether the alert is on or off.

For example:

```
04-03-19 13:33:41 \CENTDIV.$ZOOS *TANDEM.ASAP.V02 004000 ASAP ALERT
                ZOOS Node \ZERO Status Down
```

Notes

- Only nodes that have an ASAP monitor reporting to the ASAP collector will have node down alerts generated for them.
- The collector does not generate a node down alert for all nodes in an Expand network, but rather only for nodes that have an ASAP monitor on them.
- Generation of a node down alert is delayed for two minutes to be certain that Expand line handler recovery has not succeeded. This delay eliminates spurious "node down" alerts.
- An ASAP node down alert is not the same as an Expand line handler down alert generated by the ASAP SGP or Expand itself. An ASAP node down alert means that there are no available paths to the remote node, for example, it is down and all line handlers to it are down.

`OBJCOLLECT` *collect-object*

defines the Collect program object file name.

The default is `$SYSTEM.SYSTEM.ASAPCOL`.

`OBJECTIVESAUDIT` ON | OFF

enables or disables TMF for auditing of the Objectives database. The default is OFF.

`OBJECTIVESDB` *filename*

defines the name of the Objectives database file.

When you execute the first RANK or MONITOR command on a node for a specific ASAP ID, the ASAP CI always requests the name of the objectives database from the ASAPMON process referred to by that ASAP ID. If the ASAP CI cannot communicate with ASAPMON, it reverts to the database you set in the SET OBJECTIVESDB command.

The default is `$asap-database-subvolume.DBOBJ`. Due to I/O activity in the ASAP database volume, HPE recommends that you locate the objectives database file on another disk volume.

`OBJECTIVESEVENT ON | OFF`

enables ASAP for event generation when objectives are not met for attributes where event generation is specified. The default is OFF.

`OBJECTIVESEVENTUP ON | OFF`

enables ASAP for generation of up events when alerts for which down events were generated recover to an acceptable state. The default is OFF.

`OBJECTIVESEVENTCONSOLE ON | OFF`

displays event messages on the operator console. The default is ON.

`OBJECTIVESEVENTCONSOLEUP ON | OFF`

displays up event messages on the operator console. The default is ON.

`OBJECTIVESEVENTSUBJECT ON | OFF`

ON

causes ASAP to mark the domain name token (`zasp-tnk-asap-dname`) as the subject token in ASAP event buffers.

OFF

causes ASAP to mark the ASAP component token (`zasp-tnk-asap-component`) as the subject token in ASAP event buffers.

The default is `SET OBJECTIVESEVENTSUBJECT OFF`.

`OBJECTIVESEVENTSTATE OEM-state`

indicates the state at which to generate an event and/or take an action when objectives are not met for an attribute. Allowed OEM States are:

OEM-state

is one of:

LOW	Low performance alert
MEDIUM	Medium performance alert
HIGH	High performance alert
WARNING	Warning alert

CRITICAL Critical problem alert
DOWN Down object alert

The default is CRITICAL.

Note

Escalating and de-escalating goals can create High, Warning and Critical alert states so OBJECTIVESEVENTSTATE should be set to LOW, MEDIUM or HIGH when using goals of that type if those goals create EMS events or initiate automated actions.

OBJECTIVESEVENTMAX *max-events-per-domain-per-interval*

defines the maximum number of events that ASAP should generate per domain per statistics interval. The default is 1.

OBJECTIVESRANK STANDARD | PERCENT | HISTORICAL

sets the type of analysis and ranking by ASAP when applying objectives against attribute values. The default is STANDARD.

STANDARD

ASAP reports on attributes as nonranked, up, down and critical.

PERCENT

ASAP reports on attributes using all OEM states based on the percentage the attribute was over or under its objective.

HISTORICAL

ASAP uses the PERCENT model to create an initial state for an attribute, then uses the last state for the attribute to create a more critical state for the attribute.

OBJMONITOR *monitor-object*

defines the Monitor program object file name.

The default is \$SYSTEM.SYSTEM.ASAPMON.

PARTITION *partition-desc.* | *reset-desc.* | FORMAT2 { ON | OFF }

defines or resets partitioning parameters for the ASAP database. These parameters can be set or reset individually for each entity, globally for all entities, or both.

When used with no options, SET PARTITION displays the current partitioning settings.

Note

Changes to partition settings do not automatically take effect after a SET PARTITION command. To enable changes:

1. Stop the ASAP Collector process.
2. Delete or rename the existing ASAP database.
3. Restart the Collector.

This re-creates the ASAP database using the new partition settings.

partition-desc.

used to set partitioning attributes. It has the format:

```
{entity|DEFAULT} {file-opt} {part-num|BASE|ALL} part-opt  
[, part-opt...]
```

entity

the name of the specific entity (for example CPU, DISK) to which the partitioning options apply. To set default properties for all entities, use DEFAULT. These defaults are used when an entity does not have explicit partitioning attributes configured.

file-opt

is one of:

```
{ TYPE      type-value  }  
{ REFRESH  refresh-value }
```

type-value { KEYSEQUENCED | RELATIVE }

Define the ASAP historical database type for the entity as a key-sequenced or relative file type. Use TYPE RELATIVE for higher performance on databases with longer records, for example APP (ASAPX) databases with many attributes. If RELATIVE then partitioning is not allowed. As of ASAP 3.0, the default for DBTCP is RELATIVE, all others default to TYPE KEYSEQUENCED.

refresh-value { ON | OFF }

Turns ON or OFF the REFRESH EOF attribute for the entity's database. Use REFRESH OFF for higher performance on all databases. As of ASAP 3.0, the default for DBTCP is OFF, all others are ON.

part-num

the partition number (0 through 15) to which the partitioning options apply. 0 (or BASE) indicates the base partition of the entity. To specify primary and secondary extent sizes for all partitions of the entity, use ALL.

part-opt

is one of these partition options:

{	KEY	<i>key-value</i>	}
{	SYSNO	<i>system-number</i>	}
{	PRIEXT	<i>ext-value</i>	}
{	SECEXT	<i>ext-value</i>	}
{	MAXEXTENTS	<i>maxext-value</i>	}
{	VOLUME	<i>volume-name</i>	}

key-value

a partial key value, defining the lowest key value that can reside in the partition. Its format can be:

* A string of characters enclosed in quotation marks (for example, -abc1). Use this to represent partial key values that contain only printable alpha-numeric data.

* Decimal integers representing byte values from 0 through 255, enclosed in brackets (for example, [10, 13, 129]). You can use this format to represent any partial key value, on a byte-by-byte basis.

* A mixture of quoted character strings and decimal integer byte values enclosed in brackets (for example, [10, "ab", 13, 129]). You can use this format to represent any partial key value, with the quoted strings simplifying entry of printable alpha-numeric characters.

system-number

a valid NonStop™ OS system number from 0 through 255. The purpose of this value is to simplify specifying partial key values that utilize the system number portion of the ASAP record key. This value, if supplied, is prefixed to any supplied *key-value* to define the actual key used to partition the database.

ext-value

an integer value from 1 through 65535 for Format 1 files, or 1 through 512,000,000 for Format 2 files. It specifies the number of pages in the

extent. This value can be rounded up to an even number when the partition is created.

maxext-value

an integer value from 16 through 978, that specifies the maximum number of extents in the partition.

volume-name

the name of the NonStop™ OS volume on which the partition will reside.

reset-desc.

RESET

resets all partitioning attributes for all entities, including the default settings.

{ *entity* | DEFAULT } RESET

resets all partitioning attributes for the specified entity, or the default settings if you specify DEFAULT.

{ *entity* | DEFAULT } { *part-num* | ALL } RESET

resets the partitioning attributes of the specified partition of the specified entity, or default settings. To reset the partitioning attributes of all partitions for the specified entity, use ALL.

{ *entity* | DEFAULT } { *part-num* | ALL } *part-opt* RESET

resets a specific partitioning attribute for the specified entity and partition number. To reset the default setting of the specific attribute, use DEFAULT. To reset all partitions of the specified entity, use ALL.

FORMAT2 { ON | OFF }

enables or disables support of Format 2 files. Format 2 support lets ASAP retain much more data by allowing the creation of much larger files and partitions.

Notes

- You cannot specify a KEY, SYSNO, or VOLUME for a base partition (that is, partition number 0). The latter is done by the SET DB command.
- You cannot specify KEY, SYSNO, and VOLUME values if the partition number is ALL.

- You must provide a KEY or SYSNO value, or both, and a VOLUME for all secondary partitions. If you do not, the partition description is marked Invalid in the output of the SET PARTITION command, and the Collector does not create the partition.
- You must define partitioning keys (made up of a combination of the SYSNO and KEY values) in ascending order for each partition. That is, the partitioning key for partition 2 must be greater than the partitioning key for partition 1, the key for partition 3 must be greater than the key for partition 2, and so on.
- If Format 2 is on, you cannot specify MAXEXTENTS for a secondary partition. The collector ignores any such values, and the secondary partitions are created with the MAXEXTENTS value specified for the base partition.
- The PRIEXT, SECEXT, and MAXEXTENTS values specified for partitions of the DEFAULT entity are used when creating partitions for all other entities, unless you supply specific values when you define the new partitions. For example, partition 1 of the DEFAULT entity is defined with PRIEXT = 100. This value is used when creating partition 1 of the CPU entity unless the definition of partition 1 of the CPU entity overrides the DEFAULT setting.
- The Collector automatically adjusts the length of the partitioning keys specified for the partitions of a given entity to ensure that all partition key lengths are the same for that entity. This is accomplished by null-filling partition keys for the maximum partition key length specified.

For example, if the KEY for partition 1 of the APP entity is defined as FT, the KEY for partition 2 is SWIFT, and the key for partition 3 is TXN, the maximum partitioning key length is 5 bytes (from partition 2, for SWIFT). Therefore, the Collector appends null bytes to the partitioning keys for partitions 1 and 3, so their lengths are also 5 bytes. The partitioning key for partition 1 becomes [FT, 0, 0, 0], and the partitioning key for partition 3 becomes [TXN, 0, 0]. This occurs in the Collector when the database is created, so the keys are not shown in this adjusted form in the ASAP CI.

The SYSNO value is considered to be part of the key and, if specified, is factored into the key length.

- Issuing a SET DB command resets all partitioning attributes.
- The Collector uses only valid partition descriptions (those listed as Valid in the SET PARTITION command output) when creating the database. For any entity, the Collector uses all valid partition descriptions found,

beginning at partition 0, until it encounters an invalid description. All further partition descriptions for that entity are ignored.

For example, if partitions 0, 1, 2, and 4 are valid for the CPU entity, but partition 3 is invalid, only partition descriptions 0, 1, and 2 are used when creating the CPU database. Partition description 4 is ignored, even though it is valid, because description 3 is invalid.

- If partitioning is enabled for an entity, the Valid column of the SET PARTITION output indicates the status of each configured partition with one of these values:

Value	Indicates That the Partition Definition Is...	Used by Collector to Create Database?
Yes	Valid.	Yes
No-Vol	Invalid because no volume was specified.	No
No-Sys/Key	Invalid because no KEY or SYSNO was specified.	No
No-Vol,Sys/Key	Invalid because no VOLUME and no KEY or SYSNO was specified	No
No-Prior Inv	Invalid because a prior partition definition was invalid. The definition itself is correct, but a previous partition definition for the entity is incomplete.	No
??-Key Seq	Questionable because the partitioning key (a combination of the SYSNO and KEY values) is less than the partitioning key of a prior partition definition. Fix the key sequencing.	Yes, with errors
??-Dup Vol	Questionable because a prior partition definition uses the same volume. Each partition must be on a separate volume.	Yes, with errors
??-Vol Unknown	Questionable because the specified volume does not exist on the system on which ASAP is running. Either you gave a remote name for a partition volume or a configuration is being prepared on one system for another. Specify an existing volume.	Yes, with errors
??-Vol Down	Questionable because the specified volume is down. Specify a volume that is not down.	Yes, with errors
??-Partn Size	Not optimal because the maximum potential partition size defined by PRIEXT, SECEXT, and MAXEXTENTS exceeds the limit for the specified file	Yes, with errors

format.		
??-Max Ext n/a	Not optimal because the MAXEXTENTS value for the secondary partition is ignored since FORMAT2 is on. The partition is created, but the MAXEXTENTS of the base partition is used.	Yes

Examples

- To specify a default primary extent size of 256 pages and secondary extent size of 512 pages for all partitions of all files:

```
SET PARTITION DEFAULT ALL PRIEXT 256, SECEXT 512
```

- To configure three partitions for the PROCESS entity, one for system numbers 0 through 10 (base partition), one for system numbers 11 through 20 on disk \$DATA1, and one for system numbers 21 through 255 on disk \$DATA2:

```
SET PARTITION PROCESS 1 SYSNO 11, VOLUME $DATA1
```

```
SET PARTITION PROCESS 2 SYSNO 21, VOLUME $DATA2
```

- To do the same thing, but set specific extent values for partition 2:

```
SET PARTITION PROCESS 2 SYSNO 21, VOLUME $DATA2, PRIEXT 100, SECEXT 100
```

- To configure two partitions for APP data, split by system number 12 and domain "FUNDSTRANSFER", with the secondary partition on \$DATA, and with a maximum of 400 extents for each partition:

```
SET PARTITION APP 0 MAXEXTENTS 400
```

```
SET PARTITION APP 1 SYSNO 12, KEY "FUNDSTRANSFER", VOLUME $DATA
```

- To reset all partitioning-related values to their defaults:

```
SET PARTITION RESET
```

- To reset all partitioning-related values for the CPU entity to their defaults:

```
SET PARTITION CPU RESET
```

or

```
SET PARTITION CPU ALL RESET
```

- To reset the KEY value for partition 1 of the DISK entity to its default:

SET PARTITION DISK 1 KEY RESET

- To enable FORMAT2 support:

SET PARTITION FORMAT2 ON

PRIMARY *cpu* | *

defines the primary CPU number for the Collect or Monitor process started by the STARTUP command. If this CPU is unavailable, no other CPU is used. The default is asterisk (*), which is all CPUs for the specified system.

PRIORITY *server-priority*

defines the execution priority of the Monitor or Collect process started by the STARTUP command.

The default is 160.

PROCESS *entity-option*

defines startup options for Process monitoring within ASAP.

CPU *cpu-num*

defines the CPU that the PROCESS SGP process will reside in.

ON | OFF

turns Process monitoring on and off. The default is ON.

OBJECT *filename*

specifies the name of the Process SGP program file. The default is \$SYSTEM.SYSTEM.ASAPPRO.

The SET Process parameters are:

AGG

when present, ASAP creates aggregate and/or state propagated group records for processes defined as a part of logical process groups.

AGGONLY

when present, ASAP creates aggregate and/or state propagated group records for processes defined as a part of logical process groups, and it writes only these group records to the ASAP database.

MINSTATE

defines the state at which the Process SGP should consider a domain to be in an alert state. Valid values are 2 through 8; the default is 3.

MINONLY

when a MINONLY state is specified, the Process SGP operates in exception mode, writing only alerting detail domain records to the database. MINONLY is used to determine the state at which a domain is considered to be alerting and ASAP writes only those domain records to the database, removing them after they return to normal. A domain is considered to be alerting if ANY attribute state is greater than or equal to MINONLY. MINONLY is not applied to aggregate domains.

NORECS

in some rare cases no process history is required, and users do not want to monitor processes using ASAP Client. For these cases, a user can specify NORECS to cause the Process SGP to stop writing all detail process records to the database. The SGP still monitors the processes and generates EMS events (if enabled) when they alert, but it does not write records to the ASAP database for viewing by the ASAP Client or for historical tracking.

TRACE

turns \$Receive tracing on in the Process SGP. The SGP generates a log message for each read or reply.

RATE #mins

sets a specific rate for the Process SGP.

MAXAGGS

defines the maximum number of aggregate domains. The Process SGP will allocate memory to support that number of aggregate domains. Valid values are 2 through 32767. ASAP will allocate at least 1 less domain than you specify. The default setting can vary by ASAP version.

MAXDOMAINS

defines the maximum number of monitored domains. The Process SGP will allocate memory to support that number of monitored domains. Valid values are 2 through 32767. ASAP will allocate at least 1 less domain than you specify. The default setting can vary by ASAP version.

MBYTES

defines the size in megabytes of the memory pool allocated to store domains that are dynamically resolved at each interval. The default setting can vary by ASAP version.

Note

The MONITOR PROCESS, LIST, DETAIL command will return information about the maximum value parameters and the memory pool utilization.

Examples

To turn process monitoring on and to set the state at which ASAP considers a process to be alerting enter:

```
SET PROCESS ON, PARAM "MINSTATE 6"
```

To turn on aggregation for all levels of all hierarchically named processes:

```
SET PROCESS PARAM "AGG"
```

```
PROCESSBUSY ENTRIES count | SUBSAMPLES count
```

controls the collection of process statistics.

```
ENTRIES count
```

optionally specifies the number (1 through 20) of busy processes to collect or display from each CPU during a sample interval. The default is 3.

```
SUBSAMPLES count
```

optionally specifies the number (1 through 10) of subsamples to take during a sample interval. The default is 3.

The ASAP Monitor takes subsamples during one sample interval specified by the SET RATE command. Because of a potentially long sample interval, some processes might not run during the entire sample period. This makes subsampling a requirement. For example, if a sample interval is 15 minutes (SET RATE 15) and the subsample count is equal to 3, ASAP takes an internal data sample every 5 minutes (15/3). However, the final %busy CPU time of the busiest processes must be adjusted by the actual CPU process time during the sample interval (15 minutes in this example). Process statistics collected during the subsampling are not forwarded to the ASAP Collector or the database; subsample statistics are only available within an active ASAP SGP process.

Examples

This example shows the SET PROCESS command with its options:

```
+ SET PROCESSBUSY ENTRIES 5, SUBSAMPLES 3
```

PROXY ON|OFF

ON|OFF

The SET PROXY ON command enables ASAP Hybrid within ASAP Server. Hybrid settings are not visible within ASAP until this command is executed. ASAP Hybrid is an optional ASAP product.

For information about configuring the ASAP Hybrid product, see the *ASAP Hybrid Manual*.

Examples

```
SET PROXY ON
```

PROXYCONFIG *filename*

filename

The fully qualified, local ASAP Hybrid Proxy server configuration *filename*. The default is \$SYSTEM.SYSTEM.ASAPPCNF. ASAP Hybrid is an optional ASAP product.

Examples

```
SET PROXYCONFIG $DATA.ASAP.ASAPPCNF
```

PROXYCPU *number*

number

The processor where ASAP will start the ASAP Hybrid Proxy server. ASAP Hybrid is an optional ASAP product.

Examples

```
SET PROXYCPU 3
```

PROXYOBJECT *object-filename*

object-filename

The fully-qualified local filename of the ASAP Hybrid Proxy server. The default is SYSTEM.SYSTEM.ASAPPHY. ASAP Hybrid is an optional ASAP product.

Examples

```
SET PROXYOBJECT $SYSTEM.ASAP.ASAPPHY
```

PROXYPARAM *proxy-parameter*

proxy-parameter

A startup parameter for the ASAP Hybrid Proxy server. This item is for future use. ASAP Hybrid is an optional ASAP product.

RATE *minutes*

defines the interval, in minutes, that ASAP entity availability information are reported back to the process defined by the Collect and ID options. The minimum recommended sample rates for a single Collect process in a large network are:

- 1 minute for a 25-node network with 2 CPUs per node
- 2 minutes for a 25-node network with 4 CPUs per node
- 3 minutes for a 50-node network with 2 CPUs per node
- 5 minutes for a 100-node network with 4 CPUs per node
- 10 minutes for a 100-node network with 8 CPUs per node

Using multiple ASAPCOL processes and partitioning the database can significantly improve these throughput rates.

The default is 5 minutes.

RDF *entity-option*

defines startup options for RDF availability monitoring within ASAP.

CPU *cpu-num*

defines the CPU that the FILE SGP process will reside in.

ON | OFF

turns RDF monitoring on and off. The default is ON.

OBJECT *filename*

specifies the name of the Process SGP program file. The default is \$SYSTEM.SYSTEM.ASAPRDF.

PARAM "RATE #mins"

overwrites the ASAP default rate parameter for the RDF SGP only. Instructs the RDF SGP to sample at this rate.

PARAM "TRACE"

turns \$Recieve tracing on in the Process SGP. The SGP generates a log message for each read or reply.

REMOTECURITY ON | OFF

defines whether remote passwords are checked when shutting down a remote ASAP monitored node or when obtaining status information about a remote ASAP monitored node. SET REMOTECURITY OFF bypasses RemotePassword security checks. SET REMOTECURITY ON forces RemotePassword checking of any remote node. The default is ON.

RETAIN *option*

defines the action to be performed against the database at cleantime.

ALL

retains all records indefinitely. The files eventually become full unless you take manual action, such as doing a manual FUP PURGEDATA.

ROLLOVER

renames file *Zmmdxxxx*, where *mmdd* is the month and day when the files are renamed and *xxx* is the generic file name of the database file. ROLLOVER is the default RETAIN option. Any active ASAP CI process that is executing while the database rollover occurs switches to the latest ASAP database files after some user interaction occurs such as pressing the Return key (CR LF).

NONE

purges all data in the existing database at cleantime.

number

defines the number of most recent database records to keep.

The default is NONE.

Examples

This example keeps ASAP database files for historic purposes:

```
+SET RETAIN ROLLOVER
```

```
+STARTUP COLLECT
```

The RETAIN number option can have a negative performance impact in networks that monitor large numbers of nodes in a database. This occurs when it becomes necessary to process large numbers of statistics records in the database.

SLEEPTIME *time, time*

defines the time range (*hhmm, hhmm*) when statistics collection does not occur. Although statistics collection ceases, the ASAP system remains active. Shutting

down Collect and restarting it with a different sleep time reactivates collection to the database.

time, time

defines the beginning and ending times of the time range.

The default is 0000,0000; it disables sleeping.

Examples

This example sets the sleeptime:

```
+SET SLEEPTIME 0000, 0300      !No database collection
                                between Midnight and 3:00 AM
                                daily occurrence.
+STARTUP COLLECT
```

SPOOLER *entity-option* [, *entity-option*]

determines settings and parameters for the Spooler entity.

The default is OFF, \$SYSTEM.SYSTEM.ASAPSPL.

CPU *cpu-num*

defines the CPU that the SPOOLER SGP process will reside in.

ON | OFF

enables or disables network-wide spooler availability reporting.

OBJECT *ASAP-spooler-program-filename*

specifies the name of the Spooler SGP program file. The default is \$SYSTEM.SYSTEM.ASAPSPL.

PARAMETERS

specifies spooler-specific parameter string to pass to each Spooler SGP at startup.

DETAIL

instructs the Spooler SGP to return individual detail records by default. The Spooler SGP otherwise operates in aggregation mode by default, returning one aggregate record for all collectors, one for all devices, and one for all print processes.

PLUS

the PLUS option turns on high performance enhancements for Spooler environments with a large number of jobs. When PLUS is specified ASAP does not analyze and count each job in the Spooler. This reduces resource consumption dramatically but does have some limitations:

- Job detail information is reported at the Supervisor level only, it is not broken down for each Spooler component.
- Jobs in a HOLD state are not reported, the value for Hold is always zero.
- The value for the Batch and UBatch attributes will always be the same value, the number of unique batch jobs.

Use of the PLUS option is recommended when there are thousands of jobs in a Spooler subsystem.

RATE *#mins*

overwrites the ASAP default rate parameter for the Spooler SGP only. Instructs the Spooler SGP to sample at this rate.

TRACE

instructs the Spooler SGP to log trace records to the ASAP log file.

Examples

```
+SET SPOOLER ON, OBJECT $SYSTEM.SYSTEM.ASAPSPL, PARAMETERS  
"RATE 5 PLUS"
```

```
+SET SPOOLER ON, PARAMETERS "DETAIL"
```

SWAP *entity-option* [, *entity-option*]

determines settings and parameters for the Swap entity.

The default is OFF, \$SYSTEM.SYSTEM.ASAPSWP.

CPU *cpu-num*

defines the CPU that the SWAP SGP process will reside in.

ON | OFF

enables or disables network-wide swap availability reporting.

OBJECT *ASAP-swap-program-filename*

specifies the name of the Swap SGP program file. The default is \$SYSTEM.SYSTEM.ASAPSWP.

PARAMETERS

specifies swap-specific parameter string to pass to each Swap SGP at startup.

RATE *#mins*

overwrites the ASAP default rate parameter for the Swap SGP only. Instructs the Swap SGP to sample at this rate.

Examples

```
+SET SWAP ON, CPU 3, OBJECT $SYSTEM.SYSTEM.ASAPSWP, PARAMETERS  
"RATE 5"
```

SYNC *minutes, frequency*

specifies the interval (in minutes past midnight) and the number of times (frequency) that the system clock of the node in which the Monitor resides is automatically synchronized to the system clock of the node specified with the Collect option.

If Monitor and Collect are in the same node, this option has no meaning. For this SYNC option to function, the Monitor must be accessed by someone with a super-group user ID. For more information, see SYNC SLOW ON | OFF.

minutes

specifies the number of minutes past midnight.

The default number of minutes is 0.

frequency

specifies the number of times the Monitor's system clock is automatically synchronized to the node's system clock. The default frequency is 0.

The default is 0,0.

Examples

This example synchronizes all the nodes in the network twice a day, first at 2:30 a.m. and again at 5:00 a.m.

```
+SET SYNC 150, 2
```

```
+STARTUP \*
```

```
SYNCSLOW ON | OFF
```

defines the clock synchronization algorithm. The SET SYNC option controls when time-of-day synchronization occurs. The SET SYNCSLOW option controls the algorithm. You can manually correct the time on any node with the TACL SETTIME command, regardless of how SET SYNC or SYNCSLOW is set.

ON

adjusts the time-of-day clock on remote ASAP nodes very slowly to avoid sudden time of day changes. This algorithm slowly adjusts the time of day regardless of the time difference between the correct time on the central ASAP Collect node and the incorrect time on the Remote node.

OFF

causes the ASAP clock synchronization algorithm to make instantaneous time changes if the time difference between the central Collect node and the Remote node is greater than two minutes. Uses slow synchronization only if the time difference between the central Collect node and the Remote node is less than two minutes.

The default is OFF

```
TAPE entity-option [, entity-option]
```

determines the settings and parameters for the Tape entity.

The default is OFF, \$SYSTEM.SYSTEM.ASAPTAP.

```
CPU cpu-num
```

defines the CPU that the TAPE SGP process will reside in.

```
ON | OFF
```

enables or disables network-wide tape availability reporting.

```
OBJECT ASAP-tape-program-filename
```

specifies the name of the Tape SGP file. The default is \$SYSTEM.SYSTEM.ASAPTAP.

PARAMETERS

specifies tape-specific parameter string to pass to each Tape SGP at startup.

RATE #mins

overwrites the ASAP default rate parameter for the Tape SGP only.
Instructs the Tape SGP to sample at this rate.

TRACE

instructs the Tape SGP to log trace records to the ASAP log file.

Examples

```
+SET TAPE ON, OBJECT $SYSTEM.SYSTEM.ASAPTAP, PARAMETERS "RATE  
5"
```

TERM logterm | OFF

defines the Collect or Monitor terminal log-file name to which internal errors and actions are logged. The default is \$0. A disk error log file \$SYSTEM.ASAP.ASAPLOG is always created and maintained.

logterm

specifies another name for the terminal log file.

OFF

disables terminal (or \$0) logging.

ON

enables terminal (or \$0) logging.

TCP entity-option

defines startup options for TCP/IP and Telserv monitoring within ASAP.

ON | OFF

turns TCP/IP and Telserv monitoring on and off. The default is OFF.

OBJECT filename

specifies the name of the TCP SGP program file. The default is \$SYSTEM.SYSTEM.ASAPTCP.

PARAMETERS "TCP-param [TCP-param]... " | null

specifies the TCP specific parameter string to be passed to the ASAPTCP process at startup time. The default is the *<null>* string.

The Parameter list must be enclosed in double quotes.

TCP-param options are:

TRACE

turns on \$RECEIVE trace to the EMS and ASAP log files.

RATE *minutes*

defines the interval, in minutes, that TCP statistics are reported back to the Collector. The default is the ASAP RATE. The range is from 1 to 120 minutes.

SCPNAME *\$name*

defines the SCP process name the SGP will use to retrieve statistics from the TCP/IP and Telserv subsystems. If no SCPNAME is specified, the SGP will attempt to start one but security concerns could prevent a successful start of an SCP process.

TCP *n*

defines the minimum number of minutes the SGP will wait before reconfiguring TCP processes when they are being automatically configured by the SGP. The default is 60 minutes.

ROUTE *n*

defines the minimum number of minutes the SGP will wait before reconfiguring TCP Routes when they are being automatically configured by the SGP. The default is 60 minutes.

SUBNET *n*

defines the minimum number of minutes the SGP will wait before reconfiguring TCP subnets when they are being automatically configured by the SGP. The default is 60 minutes.

TELNET *n*

defines the minimum number of minutes the SGP will wait before reconfiguring Telserv processes when they are being automatically configured by the SGP. The default is 60 minutes.

PORT [*n*]

turns on monitoring of TCP Ports, and optionally specifies the minimum number of minutes the SGP will wait before reconfiguring TCP ports when they are being automatically configured by the SGP. Port monitoring is off by default, and the configuration interval defaults to the ASAP rate.

SERVICE [*n*]

turns on monitoring of Telserv Services, and optionally specifies the minimum number of minutes the SGP will wait before reconfiguring Telserv Services when they are being automatically configured by the SGP. Service monitoring is off by default, and the configuration interval defaults to 60 minutes.

WINDOW [*n*]

turns on monitoring of Telserv Widows, and optionally specifies the minimum number of minutes the SGP will wait before reconfiguring Telserv Windows when they are being automatically configured by the SGP. Window monitoring is off by default, and the configuration interval defaults to the ASAP rate.

MAXTCP *n*

defines the maximum of number of TCPPROCESS domain records that can be processed by the SGP. Domains are either autoconfigured or entered using the MONITOR command. The default is 1000 domains. The maximum is 32767.

MAXRTE *n*

defines the maximum of number of TCPROUTE domain records that can be processed by the SGP. Domains are either autoconfigured or entered using the MONITOR command. The default is 1000 domains. The maximum is 32767.

MAXSNT *n*

defines the maximum of number of TCPSUBNET domain records that can be processed by the SGP. Domains are either autoconfigured or entered using the MONITOR command. The default is 1000 domains. The maximum is 32767.

MAXPRT *n*

defines the maximum of number of TCPPORT domain records that can be processed by the SGP. Domains are either autoconfigured or

entered using the MONITOR command. The default is 1000 domains. The maximum is 32767.

MAXTEL *n*

defines the maximum of number of TELSERVPROCESS domain records that can be processed by the SGP. Domains are either autoconfigured or entered using the MONITOR command. The default is 1000 domains. The maximum is 32767.

MAXSRV *n*

defines the maximum of number of TELSERVSERVICE domain records that can be processed by the SGP. Domains are either autoconfigured or entered using the MONITOR command. The default is 1000 domains. The maximum is 32767.

MAXWIN *n*

defines the maximum of number of TELSERVWINDOW domain records that can be processed by the SGP. Domains are either autoconfigured or entered using the MONITOR command. The default is 1000 domains. The maximum is 32767.

The ASAP monitor does not check the contents of the parameter string. All parameter data is passed to the ASAPTCP process without modification.

To set PARAMETERS to null, enter:

```
SET TCP PARAMETER
```

CPU *cpu-num*

defines the CPU that the TCP SGP process will reside in.

Examples

```
SET TCP OBJECT $SYSTEM.ASAP.ASAPTCP, PARAMETERS "SCPNAME  
ZNET", CPU 2
```

```
SET TCP PARAM "SCPNAME $ZNET, PORT, SERVICE, WINDOW"
```

TMF *entity-option*

defines the settings and parameters for the TMF entity.

entity-option is one of:

CPU *cpu-num*

defines the CPU that the TMF SGP process will reside in.

ON | OFF

enables or disables network-wide TMF availability reporting. The default value is OFF.

OBJECT *ASAP-TMF-program-file-name*

defines the *ASAP-TMF program object filename*; this program is automatically started on each node by the ASAP monitor. The default is \$SYSTEM.SYSTEM.ASAPTMF.

PARAMETERS "*TMF-param* [*TMF-param*]... " | null

specifies the TMF specific parameter string to be passed to the ASAPTMF process at startup. The default is the null string.

The Parameter list must be enclosed in double quotes.

The *TMF-param* option is:

RATE *minutes*

defines the interval, in minutes, that TMF statistics are reported back to the Collector. ASAPTMF uses the interval provided by the operator. The default is the ASAP RATE.

The ASAP monitor does not check the contents of the parameter string. All parameter data is passed to the ASAPTMF process without modification.

Examples

```
+SET TMF ON, OBJECT $SYSTEM.ASAP.ASAPTMF, PARAMETERS "RATE 10"
```

To set PARAMETERS to null, enter either of:

```
SET TMF PARAMETERS
```

```
SET TMF PARAMETERS, entity-option
```

TRACE ON | OFF

turns an internal trace on and off for debugging purposes only. Use this option only when assisting the HPE Global Customer Support Center in problem diagnosis.

The default is OFF.

SHOW Command

The SHOW command displays information about the Entities defined to ASAP using the ASAP Entity Definition Language (EDL).

```
SHOW [entity] [attribute]
      [, DETAIL]
      [, ATTRIBUTES]
```

Note

Attributes that support objectives are identified with an asterisk in SHOW command output.

entity

is the name of an ASAP entity as defined in EDL.

attribute

is the name of an attribute as defined for the entity in EDL.

DETAIL

shows detailed information about the entity.

ATTRIBUTES

shows the attributes defined for an entity.

Examples

```
SHOW
SHOW DISK
SHOW DISK, DETAIL
SHOW DISK, ATTRIBUTES
SHOW FILE FULL
SHOW FILE, DETAIL, ATTRIBUTES
```

SHUTDOWN Command

The SHUTDOWN command stops ASAP Monitor, SGPs, and Collect processes on the nodes that you specify. Only someone with a super-group user ID can use this command.

```
SHUTDOWN [ \* ] [ COLLECT | MONITOR ]
          [ \node-name ] [ [.] ppd-name ]
```

*

shuts down the ASAP Monitor or Collect process on all nodes of the network.

COLLECT

specifies a Collect process.

MONITOR

specifies a Monitor process. This is the default. If SET DISK ON was specified at startup, shutdown of the Monitor process causes the DISK to be shut down as well.

node-name

specifies the individual system for which information is to be monitored or collected. It stops the ASAP Monitor or Collect process on this node only.

ppd-name

specifies a process pair directory name. The process pair directory is a list kept by the NonStop™ OS operating system of the process names and corresponding pairs of process numbers that make up each process.

Examples

These examples show the SHUTDOWN command:

```
+ SHUTDOWN \MEMPHIS           !shutdown Memphis Monitor and SGPs if running.
+ SHUTDOWN \*                  !shutdown all Monitors and SGPs if running.
+ SHUTDOWN \CHICAGO COLLECT    !shutdown Chicago Collector
```

Considerations

- SHUTDOWN shuts down Monitor (default) or Collect processes.
- After shutdown is complete, data collection stops.

SPOOLER Command

The SPOOLER command displays Spooler availability information for the specified nodes and Spooler domains. For an explanation of Spooler statistics, enter HELP SPOOLERSTATS at an ASAP prompt.

```
SPOOLER [/OUT file/] [\*] [ [.] * ] [ , SAMPLES count ]
          [\node] [ [.] spoolerspec ] [ , TIME hh[:mm [m/d/y] ]
                                     [ , STATES ]
                                     [ , DETAIL ]
                                     [ , SHORT ]
                                     [ , MINSTATE [state ] ]
```

	[AUTO]]
	[AUTOSHORT]]
[, EXACT]

OUT *file*

specifies the spooler or line printer where the ASAP output is to be sent. When no output file is specified, the output is displayed on your terminal or workstation.

*

specifies that the performance and state information for all systems is displayed.

node

specifies the individual system for which availability information is displayed. If you want information for all the systems, specify SPOOLER *. If you do not specify a node or the wild-card character (*), the default is used. The default is the system where the ASAP program was started or the system specified in the ASAP SYSTEM command.

spoolerspec

specifies only spooler records related to that spooler domain name.

*

specifies to display the performance and state information for all spoolers on the specified system. If you do not specify a spooler or the wild-card character (*), statistics for all spoolers on the current system are displayed.

SAMPLES *count*

specifies the number of sample periods to be displayed. A sample period is defined by the RATE option in the SET command. You can specify 0 through 255 sample periods. The default sample period number is 1.

TIME *hh:mm m/d/y*

defines the ending time of the display information in hours, minutes, month, day, and year. For example, specifying -TIME 10:15|| when RATE is 5 indicates that statistics collected from 10:10 to 10:15 are to be displayed.

When a -?|| appears immediately after the timestamp, it indicates that the statistics are questionable and that there was a problem obtaining the statistics. Possible reasons for questionable statistics include:

- °□ Down disk drive or down volume
- °□ Disk is being revived, formatted, or mounted

When a `-<ll` appears immediately after the timestamp, the statistics were received late by the Collector process. Late statistics indicate busy Expand lines.

hh

is the hour.

mm

is the minute.

m/d/y

is the month, day, and year. Valid *m/d/y* entries are *mm/dd/yy*, *yyyy/mm/dd*, or *mm/dd/yyyy*.

STATES

shows states associated to spooler attributes.

DETAIL

shows detailed information for the specified spooler domain (supervisor, collector, device, print process). The specified domain name indicates which type of information is returned (see [Examples](#)).

Use this display only if your spooler configuration is not too large, or if you specify only certain spooler components using the EXACT option.

SHORT

when used with DETAIL or MINSTATE, displays information in short form.

MINSTATE

shows detailed information on subdomains with states greater than or equal to what is specified.

states

indicates the minimum state to report.

AUTO

indicates to use state 2 as the minimum. This is the default setting.

AUTOSHORT

indicates to return aggregate information where state is greater than or equal to two.

EXACT

indicates to return information about domains that exactly match *spoolerspec*.

Consideration

- You cannot use TIME, SAMPLES, or * with DETAIL or MINSTATE.

Examples

- To display a list of spooler domains being monitored and a few attributes:

+ SPOOLER

It displays:

\ASAP	Spooler Domains	Status	Date	Time	NumObjs	Jobs	Full	Error
\$SPLS\COLL		Active	3/23	8:07	2/2	378	30	0
\$SPLS\DEV		Waiting	3/23	8:07	3/3	0	-	0
\$SPLS\PRINT		Dormant	3/23	8:07	4/4	0	-	0
\$SPLS\SUPER		Active	3/23	8:07	1/1	378	50	0

- To retrieve all information about the spooler collections:

+ SPOOLER, RAW

It displays:

VersionStr	Sys	Domain Name	Status	Date	Time	V	Error	NumObjs
\ASAP	1	\$SPLS\COLL	Active	2000/03/23	8:07	1	0	2/---2
\ASAP	1	\$SPLS\DEV	Waiting	2000/03/23	8:07	1	0	3/---3
\ASAP	1	\$SPLS\PRINT	Dormant	2000/03/23	8:07	1	0	2/---2
\ASAP	1	\$SPLS\SUPER	Active	2000/03/23	8:07	1	0	1/---1

continued...

#Jobs	#Batch	#Open	#Print	#Hold	#DevQs	#Locs	#Fonts	Full%
392	0	0	-	50	-	-	-	30
0	0	0	0	0	0	8	6	-
0	0	0	0	0	0	17	-	-
392	0	0	0	50	200	53	6	50

- To retrieve detailed information for multiple domain types in a spooler collection:

+ SPOOLER \$SPLS, DETAIL

It displays:

\ASAP	Spooler Domain	Status	Date	Time	Error	Cpu	BCpu	Prty
\$SPLS\COLL\	\$S	Active	03/23	8:07	0	0	1	149

```

$SPLS\COLL\$$L           Active      03/23  8:07      0  1  0  149
$SPLS\DEV\$$P275A       Waiting   03/23  8:07      0  -  -  -
$SPLS\DEV\$$P275B       Waiting   03/23  8:07      0  -  -  -
$SPLS\DEV\CENTREG.$A.#S Waiting   03/23  8:07      0  -  -  -
$SPLS\PRINT\$$SPLP      Dormant   03/23  8:07      0  3  2  130
$SPLS\PRINT\$$FASTP     Dormant   03/23  8:07      0  1  2  130
$SPLS\SUPER\$$SPLS      Active    03/23  8:07      0  1  2  149

```

continued...

```

Jobs Batch  Open Print  Hold DevQs  Locs Full% File/Process Name
-----
378      0      0      0    100      -      -    75 $DATA2.SPL.DATA
14      0      0      0      0      -      -    40 $DATA2.SPL.DATA2
0       0      0      0      0      0      6     - $FASTP
0       0      0      0      0      0      6     - $FASTP
0       0      0      0      0      0      0     - $$SPLP
0       0      0      0    100      0     12     - $$SYSTEM.FASTP.FASTPTCP
0       0      0      0      0      0      0     - $$SYSTEM.SYSTEM.FASTP
392      0      0      0    100    200    53    50 $0

```

To retrieve detailed information about the spooler supervisor \$SPLS:

```
+ SPOOLER $SPLS\SUPER\$$SPLS, DETAIL
```

It displays:

```

\ASAP Spooler Supervisor      Status      Date  Time  Error Cpu BCpu Prty
-----
$SPLS\SUPER\$$SPLS           Active      03/23  8:07      0  1  2  149

```

continued...

```

Jobs Batch  Open Print  Hold Fonts DevQs  Locs Colls  Devs PProcs Full%
-----
392      0      0      0    100      6    200    53    2    6    3    50

```

continued...

LogFile

```
-----
$0
```

To retrieve detailed information about the spooler collector \$SPLS\COLL\\$\$S:

```
+ SPOOLER $SPLS\COLL\$$S, DETAIL
```

It displays:

```

\ASAP Spooler Collectors      Status      Date  Time  Error Cpu BCpu Prty
-----
$SPLS\COLL\$$S               Active      03/23  8:07      0  0  1  149

```

continued...

Data File Name	Jobs	Batch	Open	Print	Hold	Size	UnitAlloc
\$DATA2.SPL.DATA	378	0	0	0	100	4	1916

continued...

TotUnits	Full%
11998	15

- To retrieve detailed information about the spooler device \$SPLS\DEV\SP257A:
+ SPOOLER \$SPLS\DEV\SP257A, DETAIL

It displays:

\ASAP Spooler Devices	Status	Date	Time	Error	PrtProc	Flags
\$SPLS\DEV\SP257A	Waiting	03/23	8:07	0	\$FASTP	

continued...

Jobs	Batch	Hold	Locs	Fonts	DevQs	Retry	Intvl	RStrt	Max	FormName
0	0	0	6	6	0	0	360	120	5	

- To retrieve detailed information about the spooler print process \$SPLS\PRINT\FASTP:
+ SPOOLER \$SPLS\PRINT\FASTP, DETAIL

It displays:

\ASAP Spooler PrintProcess	Status	Date	Time	Error	Cpu	BCpu	Prty	Flags
\$SPLS\PRINT\FASTP	Dormant	03/23	8:07	0	3	2	130	

continued...

Program File Name	Jobs	Batch	Print	Hold	Locs	Devs	DevQs
\$SYSTEM.FASTP.FASTPTCP	0	0	0	0	17	2	0

STARTUP Command

The STARTUP command starts ASAP Monitor, SGPs, and Collect processes on the nodes that you specify. Only someone with a super-group user ID can use this command.

STARTUP	[*]	[COLLECT MONITOR]
	[\node-name]	

*

starts the ASAP Monitor or Collect process on all nodes of the network.

COLLECT

starts the ASAP object specified by the SET OBJCOLLECT option.

MONITOR

starts the ASAP object specified by the SET OBJMONITOR option. If SET *SGPName* ON is specified, the *SGPName* specified by SET *SGPName* OBJECT is started. SGP options are sent to each remote monitor. If different SGP options are required for different SGPs, each SGP must have the options set before starting the remote monitor. For example, if Disk performance statistics and operational state information are required for NODEA at an interval of 10 minutes but not for NODEB, and the Disk interval is the same as the Monitors' interval (5 minutes), these ASAP commands have to be entered:

```
+SET RATE 5
+SET DISK ON,PARAMETERS "RATE 10"
+START \NODEA
+SET DISK ON, PARAMETERS      !Parameters nulled
+START \NODEB
```

node-name

specifies the individual system for which information is to be monitored or collected. This starts the ASAP Monitor or Collect process on this node only. The default is Monitor.

Example

This example shows how to use the STARTUP command:

```
+ SET DB $DATA1.ASAP-DB      ! Specify Collect DB
+ STARTUP \LA COLLECT        ! Start Collect server
+ SET COLLECT \LA           ! Specify where Collect DB server resides
+ STARTUP \NEWYORK           ! Start Monitor on NEWYORK,
+ STARTUP \SANFRAN MONITOR   ! MONITOR is the default
+ STARTUP \CHICAGO           ! Start Monitor on \Chicago
```

Considerations

- The STARTUP command starts Monitor, SGPs, or Collect processes on the specified nodes with the specified options. The object file to use is specified with the SET OBJMONITOR and OBJCOLLECT options. For SGP processes, the SGP object is specified with the SET *SGPName* OBJECT option.

ACTIONS

option displays detailed action server information:

```
STATUS ACTIONS ASAP
```

shows analysis of last action request.

```
STATUS ACTREQUEST
```

shows XML parameters for last action request.

```
STATUS ACTIONS RUN
```

shows date-time-detail of actions run.

```
STATUS ACTIONS MSGS
```

shows oldest action server log messages.

```
STATUS ACTIONS MSGS <inx>
```

shows the <inx> most recent actions.

```
STATUS ACTIONS MSGS <inx> <lines>
```

shows <inx> actions for <lines>

```
STATUS ACTIONS LOG ON | DETAIL
```

turns on action logging. This option is not recommended except for debugging due to overhead.

```
STATUS ACTIONS TRACE ...
```

is for development purposes only.

Table 6-10. STATUS Command Output

Column Heading	Definition
Primry	Identifies the process primary CPU,PIN (process identification number)
Backup	Identifies the process backup CPU,PIN
Pri	Indicates the process execution priority

Access	Indicates the process NonStop™ OS operating system
Security	GROUP,USER access ID
Reqs	Indicates the number of requests this process has received since it was started
Lmsgs	Indicates the number of log messages this process has logged
Created	Indicates the month/day/hour that this process was started
ReqTime	Indicates the time in hours/minutes/seconds that this status request was received by each process
SrvTime	Indicates the time in hours/minutes/seconds that this status request was replied to by each process
Et	Indicates the elapsed time for this process to reply to the status request

The Backup column, instead of showing the backup CPU, PIN, might indicate one of:

Display	Meaning
APP	ASAP Extension
CIP	CIP SGP
CMM	Comm SGP
DSK	Disk SGP
EXP	Expand SGP
FIL	File SGP
NCP	Node SGP
PRO	Process SGP
RDF	RDF SGP
SPL	Spooler SGP
SWP	Swap SGP
TAP	Tape SGP
TCP	TCP/IP SGP
TMF	TMF SGP

Example - Status Monitor

```
+STATUS \ISLAND          ! Show \ISLAND's Status

\ISLAND   Primry Backup Pri Access  Reqs  Lmsgs Created  ReqTime  SrvTime  Et
-----
$BOOM V05  1,196  0,177 160 255,255  193    6  3/18 11 1423:46 1423:46  0
$BOOM0 V05  0,145          160 255,255  536    1  3/18 11 1423:46 1423:46  0
$BOOM1 V05  1,203          160 255,255  536    1  3/18 11 1423:46 1423:46  0
$BOOM2 V05  2,216          160 255,255  536    1  3/18 11 1423:46 1423:46  0
$BOOM3 V05  3,003          160 255,255  536    1  3/18 11 1423:46 1423:46  0
$BOON V04  0,142  RDF   160 255,255  192    1  3/18 11 1423:46 1423:46  0
$BOOG V05  1,245  NCP   160 255,255  192    1  3/18 11 1423:46 1423:46  0
$BOOJ V05  1,180  DSK   160 255,255  192    2  3/18 11 1423:46 1423:46  0
$BOOK V05  0,120  EXP   160 255,255  192    1  3/18 11 1423:46 1423:46  0
$BOOL V05  1,021  FIL   160 255,255  192    1  3/18 11 1423:46 1423:46  0
$BOOO V05  1,119  SPL   160 255,255  192    1  3/18 11 1423:46 1423:46  0
$BOOQ V05  1,117  TAP   160 255,255  192    1  3/18 11 1423:46 1423:46  0
$BOOR V05  1,246  TMF   160 255,255  192    1  3/18 11 1423:46 1423:46  0
$BOOT V05  1,103  PRO   160 255,255  192    1  3/18 11 1423:46 1423:46  0
$BOOU V05  1,208  COM   160 255,255  192    2  3/18 11 1423:46 1423:46  0
$BOOV V05  2,069  TCP   160 255,255  189    3  3/18 11 1423:46 1423:46  0
$BOOA V05  1,204  SWP   160 255,255  192    1  3/18 11 1423:46 1423:46  0
$BOOC V05  1,204  CIP   160 255,255  192    1  3/18 11 1423:46 1423:46  0
(collect \CENTDIV.$BOOS, rate 1)
```

Example - Status Collect

```
+STATUS COLLECT

\CHICAGO Primry Backup Pri Access  Reqs  Lmsgs Created  ReqTime  SrvTime  Et
-----
$ZOOS V04  1,100  0,238 160 255,255 15261 26962  3/17 19 1429:26 1429:26  0
$HIT.ASAPDB.DBAPP Open(0) Days NONE
$HIT.ASAPDB.DBCIP Open(0) Days 7
$HIT.ASAPDB.DBCOM Open(0) Days 30
$HIT.ASAPDB.DBCPU Open(0) Days 365
$HIT.ASAPDB.DBDSK Open(0) Days 7
$HIT.ASAPDB.DBEXP Open(0) Days 7
$HIT.ASAPDB.DBFIL Open(0) Days 7
$HIT.ASAPDB.DBNCP Open(0) Days 7
$HIT.ASAPDB.DBPRO Open(0) Days 7
$HIT.ASAPDB.DBPCB Open(0) Days 30
$HIT.ASAPDB.DBRDF Open(0) Days NONE
$HIT.ASAPDB.DBSPL Open(0) Days 30
$HIT.ASAPDB.DBSWP Open(0) Days 1
$HIT.ASAPDB.DBTAP Open(0) Days 7
$HIT.ASAPDB.DBTCP Open(0) Days 7
$HIT.ASAPDB.DBTMF Open(0) Days 7
(rate 1, sleep 0000,0000, clean 0000, retain ROLLOVER, DBMaxDays 1)
```

Example - Status Actions

```
STATUS \CHICAGO ACTIONS           ! Status chicago action server
STATUS \CHICAGO ACTIONS RUN       ! Status chicago of actions run
STATUS \CHICAGO ACTIONS ASAP      ! Analysis of last action request
STATUS ACTIONS MSGS 10            ! Show 10 most recent action msgs
STATUS ACTIONS MSGS 100 10       ! Show 100th thru 91st most recent msgs
STATUS ACTIONS **                 ! Show taskids for all action tasks
STATUS ACTIONS LOG ON             ! Enable detail log, will show logname
STATUS ACTIONS LOG OFF           ! Disables detail logging
```

SWAP Command

The SWAP command displays SWAP availability statistics for the specified nodes and swap file domains. Type HELP SWAPSTATS at the ASAP prompt for an explanation of SWAP performance statistics. For an alphabetical list, see the [CPU Statistics](#) table.

SWAP	[/OUT <file>/]	[*]	[,	SAMPLES <count>]
		[\<node>]	[[.] <cpu>]	[,	TIME hh[:mm [m/d/y]]]
					[,	CPU <number>
					[,	DETAIL
					[,	FILES
					[,	MB
					[,	RAW
					[,	STATE

OUT file

specifies the spooler or line printer where the ASAP output is to be sent. When no output file is specified, the output is displayed on your terminal or workstation.

specifies that the performance and state information for all systems is to be displayed.

node

specifies the individual system for which statistical information is to be displayed. If you want information for all the systems, specify SWAP *. If you do not specify a node or the wild-card character (*), the default is used. The default is the system where the ASAP program was started, or the system specified in the ASAP SYSTEM command.

cpu

specifies the individual CPU for which Swap performance and state information is to be displayed. All of the Swap and Swap files from all of the collected nodes can be output by specifying SWAP *.*. If you do not specify a CPU or the wild-card character (*), the default is used. The default is all the swap and swapfiles for all of the CPU's in the current system.

specifies that the performance and state information for all swap files on the specified system are to be displayed. If you do specify a CPU or the wild-card character (*), statistics for all swap files in the current system are displayed.

SAMPLES *count*

specifies the number of sample periods to be displayed. A sample period is defined by the RATE option in the SET command. You can specify 0 through 255 sample periods. The default sample period number is 1.

TIME *hh:mm m/d/y*

defines the ending time of the display information in hours, minutes, month, day, and year. For example, specifying -TIME 10:15|| when RATE is 5 indicates that statistics collected from 10:10 to 10:15 are to be displayed.

When a -?|| appears immediately after the timestamp, it indicates that the statistics are questionable and that there was a problem obtaining the statistics. Possible reasons for questionable statistics include:

- Swap files are down or the CPU for the swap files is down.
- The CPU is being reloaded for the swap files.
- A swap file was configured but does not physically exist.

When a -<|| appears immediately after the timestamp, the statistics were received late by the Collector process. Late statistics are an indication of busy Expand lines.

hh

is the hour.

mm

is the minute.

m/d/y

is the month, day, and year. Valid *m/d/y* entries are *mm/dd/yy*, *yyyy/mm/dd*, or *mm/dd/yyyy*.

CPU *number*

lists the statistics for swap files that are currently operating in the specified CPU number, *cpu-number*.

DETAIL

displays the processors swap files statistics as a whole for the specified CPU or system. Displays both the good and failed statistics for the following:

OK/FAILED for each CPU

- ALLOCATES The number of allocations made.
- CREATES The number of reservations made for process creations.
- FREES The number of allocations freed.
- RELEASES The number of reservations released.
- RESIZES The number of reservations resized.

FILES

displays each swap file configured for each CPU and displays the following statistics for each swap file:

- FILE NAME The file name of the swap file.
- FULL% The percentage of the file used.
- PAGES The total memory pages currently allocated for the file.
- RESERVED The total memory pages currently reserved in the file.
- AVAILABLE Total memory pages that are currently available in the file.

MB

displays usage statistics in megabytes

STATE

shows all attributes that have an associated state.

RAW

displays USE statistics in a dense tabular format rather than the standard ASAP sparse format. When used with the file option the individual swap file statistics and the additional 2 statistics will be displayed:

- MAXRSRVD The highest number of memory pages ever reserved in the file.
- THRESHOLD Number of memory pages that must be allocated to generate an NSK subsystem threshold alert message 200

SWAP Statistics

The following table lists the CPU statistics and options displayed when you enter the ASAP HELP SWAPSTATS command.

Table 6-11. SWAP Statistics and Descriptions

Statistics	Description/Option
ALLOCATES	Number of good and failed allocations made for each CPU.
AVAILABLE	The default display is the number of memory pages available for each CPU. With the file option it displays number of memory pages available for each file.
AVAIL%	Percent of swap space available for each CPU.
CREATES	Number of good and the number of failed reservations made for process creations for each CPU.
FILE NAME	Name of each swap file configured/monitored for each CPU.
FILES	Number of swap files configured for each CPU.
FREES	Number of good and the number of failed allocations freed for each CPU.
FULL%	Percentage swap file used for each swap file per CPU.
MAXRSRVD	The highest number of memory pages ever reserved in the file.
PAGES	Number of pages allocated for each swap file.
RELEASES	Number of good and the number of failed reservations released for each CPU.
RESERVED	The default display with no command options displays memory pages allocated for each CPU. With the file option it displays the number of pages reserved in each swap file.
RESERVATIONS	The average percentage CPU interrupt busy for the sample
RESIZES	Number of good and the number of failed reservations resized for each CPU.
STATUS	Current status of the swap files for the CPU.
THRESHOLD	Number of memory pages that must be allocated to generate an NSK subsystem threshold alert message 200
TIME	Time of the stats converted to data base local civil time.
SWAPTOTAL	Total swap space allocated for each CPU.

Examples

These examples show some of the available options and their output:

+swap, sample 4

\CENTDIV	Time	Status	Files	SWAPTOTAL	Avail	FULL	Reserved	Reservations	Available
					----	----	-----	-----	-----
					----	----	-----	-----	-----
-									
00	3/18 12:59	Up	3	4.4 GB	95	4	12379	265	256421
00	3/18 12:58	Up	3	4.4 GB	95	4	12379	265	256421
00	3/18 12:57	Up	3	4.4 GB	95	4	2379	265	256421
00	3/18 12:56	Up	3	4.4 GB	95	4	2379	265	256421
01	3/18 12:59	Up	2	1.9 GB	61	38	43835	382	71365
01	3/18 12:58	Up	2	1.9 GB	61	38	43835	382	71365
01	3/18 12:57	Up	2	1.9 GB	61	38	43835	382	71365
01	3/18 12:56	Up	2	1.9 GB	61	38	43835	382	71365
02	3/18 12:59	Up	2	1.2 GB	88	11	8318	166	63874
02	3/18 12:58	Up	2	1.2 GB	88	11	8318	166	63874
02	3/18 12:57	Up	2	1.2 GB	88	11	8318	166	63874
02	3/18 12:56	Up	2	1.2 GB	88	11	8318	166	63874
03	3/18 12:59	Up	2	1.2 GB	87	12	9309	166	62883
03	3/18 12:58	Up	2	1.2 GB	87	12	9309	166	62883
03	3/18 12:57	Up	2	1.2 GB	87	12	9309	166	62883
03	3/18 12:56	Up	2	1.2 GB	87	12	9309	166	62883

+Swap, File

\CENTDIV	Time	File Name	Full	Pages	Reserved	Available
				----	-----	-----
				----	-----	-----
00	3/18 13:45	\$SYSTEM.ZSYSSWAP.SWAP00X	6	102400	6324	96076
00	3/18 13:45	\$ASAP.SWAPFILE.SWAP00	2	102400	2511	99889
00	3/18 13:45	\$HIT.ZZSWAP.SWAP00	5	64000	3544	60456
01	3/18 13:45	\$SYSTEM.ZSYSSWAP.SWAP01	56	51200	29165	22035
01	3/18 13:45	\$HIT.ZZSWAP.SWAP01	22	64000	14670	49330
02	3/18 13:45	\$SYSTEM.ZSYSSWAP.SWAP02	57	8192	4723	3469
02	3/18 13:45	\$HIT.ZZSWAP.SWAP02	5	64000	3595	60405
03	3/18 13:45	\$SYSTEM.ZSYSSWAP.SWAP03	46	8192	3802	4390
03	3/18 13:45	\$HIT.ZZSWAP.SWAP03	8	64000	5525	58475

+Swap 3, Detail, samples 5, time 13:43

\CENTDIV	Time	OK/Fail	Creates	Releases	Resizes	Allocates	Frees
03	3/18 13:43	OK	7	7		3	
03	3/18 13:43	Failed					
03	3/18 13:42	OK					
03	3/18 13:42	Failed					
03	3/18 13:41	OK				1	
03	3/18 13:41	Failed					
03	3/18 13:40	OK					
03	3/18 13:40	Failed					
03	3/18 13:39	OK				1	
03	3/18 13:39	Failed					

SYNC Command

The SYNC command requests the Monitor on the specified node to set the NonStop™ OS time of day clock to the NonStop™ OS time of day clock on the node where the Collect server resides.

SYNC	[*]	[MONITOR]
	[\node-name]	[[.] ppd-name]

*

synchronizes information on all nodes.

MONITOR

synchronizes information on the Monitor process.

node-name

specifies the individual system for which information is to be displayed.

ppd-name

specifies a process pair directory name. The process pair directory is a list of the process names and corresponding pairs of process numbers that make up each process kept by the NonStop™ OS operating system.

Example

This example shows how to synchronize the Chicago Monitor's Greenwich Mean Time (GMT) to the San Francisco Collector's GMT:

```
+ SET COLLECT \SANFRAN
+ STARTUP \SANFRAN COLLECT ! San Francisco collector startup
+ STARTUP \CHICAGO MONITOR ! Chicago Monitor startup
+ SYNC \CHICAGO           ! Sync CHICAGO gmt to SANFRAN gmt
```

SYSTEM Command

The SYSTEM command specifies another default node name or number. Entering SYSTEM alone sets the system to the local system name.

SYSTEM	[\node-name]
	[\node-number]

node-name

specifies a node name.

node-number

specifies a node number.

Example

```
+ SYSTEM \MEMPHIS
```

TAPE Command

The TAPE command displays Tape availability information for the specified nodes and Tape domains. For an explanation of Tape statistics, enter HELP TAPSTATS at an ASAP prompt.

```
TAPE [/OUT file/] [\* ] [ [.] * ] [, SAMPLES count ]
      [\node] [ [.] tapespec] [, TIME hh[:mm [m/d/y]]]
                                     [, STATES ]
                                     [, DETAIL ]
                                     [, EXACT ]
```

OUT *file*

specifies the spooler or line printer where the ASAP output is to be sent. When no output file is specified, the output is displayed on your terminal or workstation.

*

indicates to retrieve all tape records from all nodes.

node

indicates to retrieve tape records only from the specified node.

tapespec

indicates to retrieve only tape records related to the specified tape device.

SAMPLES *count*

indicates the number of records to return.

TIME *hh:mm m/d/y*

indicates to retrieve information starting at the specified time.

STATES

indicates to display states associated with tape attributes.

DETAIL

indicates to display detailed tapemount information for the specified tape domain.

EXACT

indicates to retrieve only information that exactly matches *tapespec*.

Consideration

- You cannot use TIME, SAMPLES, or * with DETAIL.

Examples

- To display a list of tape domains being monitored and a few attributes:

+ TAPE

It displays:

\ASAP	Status	Date	Time	DevNo	CPU	BCPU	ACS	Checks	Mnts	Mtime	TapeName
\$TAPE0	Free	3/23	8:07	378	02	01	No	NL, BLP	0	0	-
\$DLT00	InUse	3/23	8:07	235	0	1	Yes	BLP	0	0	BK0001

continued...

ProgName

-

BACKUP

+

- To display a list of all tape information for all monitored tape domains:

+ TAPE, RAW

It displays:

Version	Sys	Domain	Status	Date	Time	V	DevNo	CPU	BCPU	Sub	ACS
\ASAP	1	\$TAPE0	Free	2000/03/23	08:07	1	378	2	1	6	No
\ASAP	1	\$DLT00	InUse	2000/03/23	08:07	1	255	0	1	9	Yes

continued...

Checks	Mnts	Mtime	TapeName	TpStatus	TpLabel	ProcName	ProgName	UserId
NL, BLP	0	0	-	-	-	-	-	-
BLP	0	0	BK001	SCRATCH	BACKUP	\ASAP.\$BKUP	BACKUP	255,23

+

To display detailed information for a tape device (\$DLT00):

+ TAPE \$DLT00, DETAIL

If no tapemount requests are pending, it displays:

\ASAP	Status	Date	Time	DevNo	CPU	BCPU	Sub	ACS	Checks	Mnts	Mtime
\$DLT00	InUse	3/23	8:07	255	00	01	9	Yes	BLP	-	0

continued...

TapeName	TpStatus	TpLabel	ProcName	ProgName	UserId	Prot	IO	Action
BK0001	SCRATCH	BACKUP	\$BKUP	BACKUP	255,23	-	-	-

+

If tapemount requests are pending, it displays:

\ASAP	Status	Date	Time	DevNo	CPU	BCPU	Sub	ACS	Checks	Mnts	Mtime
\$DLT00	Free	3/23	8:07	378	0	1	9	Yes	BLP	45	45
\$DLT00	Free	3/23	8:07	255	0	1	9	Yes	BLP	238	5

continued...

TapeName	TpStatus	TpLabel	ProcName	ProgName	UserId	Prot	IO
BK0002	6250	BACKUP	\ASAP.\$BKUP	BACKUP	255,23	No	OUT FUP005
6250 ANSI	\WINE.\$FUP1	FUP	23,23	Yes	IN		

continued...

Action

MOUNT REQUESTED TAPE
MOUNT REQUESTED TAPE

+

To display detailed information for the node (\ASAP) tape drive:

+ TAPE ASAP, DETAIL

It displays:

\ASAP	Status	Date	Time	DevNo	CPU	BCPU	Sub	ACS	Checks	Mnts	Mtime
ASAP	NotFound	3/23	8:07	-	-	-	-	-	-	100	15

continued...

TapeName	TpStatus	TpLabel	ProcName	ProgName	UserId	Prot	IO
-	-	ANSI	\ASAP.\$PROC	FUP	33,23	No	OUT

continued...

Action

MOUNT ANY APPROPRIATE SCRATCH TAPE
+

TCP Command

```
TCP [/OUT <file>/] [\* ] [ [.] * ] [,SAMPLES count ]
                    [\node] [ [.] domain][,STATES ]
                    [,TIME hh[:mm [m/d/y]]
                    [, MINSTATE [AUTO] ]
                    [ [state] ]
                    [, COUNT count ]
                    [,EXTRACT filename ]
                    [,NOPRINT ]
                    [,NOLINEBREAK ]
                    [,AGE minutes ]

entity [\* ] [ [.] * ] [, DETAIL ]
        [\node] [ [.] domain][,SAMPLES count ]
        [,STATES ]
        [,TIME hh[:mm [m/d/y]]
        [, MINSTATE [AUTO] ]
        [ [state] ]
        [, COUNT count ]
        [,EXTRACT filename ]
        [,NOPRINT ]
        [,NOLINEBREAK ]
        [,AGE minutes ]
```

The TCP command displays TCP and Telserv entity statistics for the specified nodes, entities, and domain names.

entity Any TCP or Telserv sub-entity name. Valid sub-entity names are:

TCPARP	(or ARP)
TCPICMP	(or ICMP)
TCPICMP6	(or ICMP6)
TCPIGMP	(or IGMP)
TCPIP	(or IP)
TCPIP6	(or IP6)
TCPLOOP	(or LOOP)
TCPMONGQ	(or MONGQ)
TCPPORT	(or PORT)

TCPPROCESS	(or TCP)
TCPQIO	(or QIO)
TCPROUTE	(or ROUTE)
TCPRTE	(or RTE)
TCPSOCK	(or SOCK)
TCPSUBNET	(or SUBNET)
TCPUDP	(or UDP)
TELSERVPROCESS	(or TELSERV, or TP)
TELSERVSERVICE	(or SERVICE, or TS)
TELSERVWINDOW	(or WINDOW, or TW)

domain

Any TCP or Telserv domain (object) name

SAMPLES

the number of samples to display per domain

STATES

the objective states of TCP metric values

TIME

defines ending time and date of sample(s) to display

DETAIL

displays custom detailed TCP metric values for specific sub-entities

MINSTATE

displays records based on attribute states

state

show records with at least one attribute state equal to or higher than *<state>*. Valid states are 1 through 8

AUTO

show records with at least one attribute state greater than state 2 (OK)

COUNT

defines the max number of records to display

EXTRACT

extract TCP records from the ASAP database to a separate database file or to a process. If the specified file does not exist, it will be created

NOPRINT

suppress terminal output for EXTRACT commands

NOLINEBREAK

do not break output lines at 132 characters. Be sure the output file can handle lines longer than 132 bytes before using this option

AGE

return removed objects for the past number of minutes specified

Examples

- To show the latest TCP/IP and TELSERV statistics for the default node:
TCP
- To show the latest TCP/IP statistics for all nodes:
TCP *
- To show the latest TCP/IP Route detailed statistics for the local node:
TCP TCPROUTE, DETAIL
- To show the latest detailed TCP/IP statistics for Port 23 on \Chicago:
TCP TCPPORT \CHICAGO.23, DETAIL
- To show the latest detailed TELSERV Window statistics for the last four samples:
TCP WINDOW, DE, S4
- To extract the latest TCPICMP statistics for all for 1000 samples to a file:
TCP ICMP, EXTRACT \$DATA.DATA.ICMP, NOPRINT, SAMPLES 1000
- To extract the last 1000 samples of TCPIP statistics into a CSV file:

TCP/OUT IPCSV/IP, RAW, DE, ST, TAB " ", NOLINEBREAK, SAMPLES
1000

TMF Command

The TMF command displays TMF availability information for the specified nodes and TMF domains. For an explanation of TMF statistics, enter HELP TMFSTATS at an ASAP prompt.

```
TMF [/OUT file/] [\* ] [ [.] * ][, SAMPLES count ]
      [\node] [ [.] domain][, TIME hh[:mm [m/d/y]] ]
                        [, STATES ]
                        [, DETAIL ]
                        [, MINSTATE [state ] ]
                                [AUTO ] ]
                                [AUTOSHORT] ]
```

*

indicates to retrieve all tape records from all nodes.

node

indicates to retrieve tape records only from the specified node.

domain

indicates to retrieve only tape records related to the specified domain.

SAMPLES *count*

indicates the number of samples to display.

TIME *hh:mm m/d/y*

indicates to retrieve information starting at the specified time.

hh

is the hour.

mm

is the minute.

m/d/y

is the month, day, and year. Valid *m/d/y* entries are *mm/dd/yy*, *yyyy/mm/dd*, or *mm/dd/yyyy*.

STATES

indicates to display all states associated with attributes.

DETAIL

indicates to display detailed information for the specified domain, without states.

EXACT

indicates to retrieve only information that exactly matches *tapespec*.

MINSTATE

displays TMF subdomains depending on the selected option:

state

shows subdomains with an OEM-state greater than or equal to *state*.

AUTO

shows subdomains with an OEM-state greater than 2, or else shows all subdomains.

AUTOSHORT

shows subdomains with an OEM-state greater than 2, or else shows the domain record.

Consideration

- You cannot use TIME, SAMPLES, or * with MINSTATE.

Examples

For descriptions of the fields displayed in TMF command output, enter `help tmfstats`.

- To display a list of monitored TMF domains, and a few attributes:

```
+ tmf
\ASAP TMF      Status   Date   Time   Type     Active  Used    Tps  Trans  Hung
-----
Auditdump     Enabled  3/14  13:18  Master   Yes     -      -    -    -
```



```

Audittrail      Up          3/14 13:18 Master -          62          -          - -
Tmf             Enabled    3/14 13:18 -          -          0.00        - -
Transactions    Up          3/14 13:18 -          -          -          -          0 No
+

```

To display only the attributes associated with a state:

```

+ tmf, states
\ASAP Tmf      Status   S Date   Time   Error S Active   S Ovflw S Hold S
Used
-----
Auditdump      Enabled  2  3/14 13:18   0 1 Yes    7 -    1 -    1 -
Audittrail     Up       2  3/14 13:18   0 1 -     1 No   1 No   1 62
Tmf            Started  2  3/14 13:18   0 1 -     1 -    1 -    1 -
Transactions    Up       2  3/14 13:18   0 1 -     1 -    1 -    1 -

```

continued...

```

S          Tps S BeginTr  S Trans S Hung S Duration S
-----
1          - 1 -          1 - 1 - 1 -          - 1
7          - 1 -          1 - 1 - 1 -          - 1
1          0.00 1 Enabled  1 - 1 - 1 -          - 1
1          - 1 -          1  0 1 No  1 -          0 1
+

```

To display all the available attributes without the states:

```

+ tmf, detail
\ASAP Tmf      Status   Date   Time   Error Type   Active   Ovflw Hold Used
-----
Auditdump      Enabled  3/14 13:18   0 Master Yes    -    -    -
Audittrail     Up       3/14 13:18   0 Master -     No    No    0
Tmf            Started  3/14 13:18   0 -     -     -    -    -
Transactions    Up       3/14 13:18   0 -     -     -    -    -

```

continued...

```

          Tps BeginTr  Catalog  Trans Hung Duration
-----
          - -          -          - -          -
          - -          -          - -          -
          0.00 Enabled  Active   - -          -
          - -          -          0 No    0
+

```

To display all the available attributes with the associated states:

```

+ tmf, states, detail
\ASAP Tmf      Status   S Date   Time   Error S Type   Active   S Ovflw S
-----
Auditdump      Enabled  2  3/14 13:18   0 1 Master Yes    7 -    1
Audittrail     Up       2  3/14 13:18   0 1 Master -     1 No   1
Tmf            Started  2  3/14 13:18   0 1 -     -     1 -    1
Transactions    Up       2  3/14 13:18   0 1 -     -     1 -    1

```

continued...

Hold	S	Used	S	Tps	S	BeginTr	S	Catalog	Trans	S	Hung	S	Duration	S
-	1	-	1	-	1	-	1	-	-	1	-	1	-	1
No	1	62	7	-	1	-	1	-	-	1	-	1	-	1
-	1	-	1	0.00	1	Enabled	1	Active	-	1	-	1	-	1
-	1	-	1	-	1	-	1	-	0	1	No	1	0	1

+

To display details about all TMF transactions subdomains:

+ tmf transactions, minstate 1

\ASAP Transactions	Status	S	Date	Time	Duration	S
\Asap.0.1583336	Active	2	3/14	13:18	132	1

continued...

Process	Dist	Parent	Children
\$Z356(0,245)	No	-	-

+

- To display details about all TMF audittrail subdomains where at least one attribute has a state value greater than 2 (if no audittrail subdomains meet this condition, all audittrail subdomains are displayed):

```
+ tmf audittrail, minstate auto
\ASAP AuditTrail Date Time Ovflw S Hold S Used% S Thresh%
-----
```

AuditTrail	Date	Time	Ovflw	S	Hold	S	Used%	S	Thresh%
Master	3/14	13:18	No	1	No	1	62	7	80

continued...

```
FirstPinnedFile Reason Dump TransDsbl%
-----
```

FirstPinnedFile	Reason	Dump	TransDsbl%
\$DSV.ZTMFAT.AA000042	NotDumped	No	90

- To display details about all TMF auditdump subdomains where at least one attribute has a state value greater than 2 (if no auditdump subdomains meet this condition, all auditdump subdomains are displayed):

```
+ tmf auditdump, minstate auto
\ASAP AuditDump Status Date Time Active S Process
-----
```

AuditDump	Status	Date	Time	Active	S	Process
Master	Enabled	3/14	13:18	Yes	7	\ASAP.\$X943

continued...

```
CurrentFile
-----
$DSV.ZTMFAT.AA000042
+
```

VOLUME Command

The VOLUME command displays and modifies the default volume and subvolume. The default node name is also shown.

```
VOLUME [ $ volume ] [ . subvolume ]
```

\$ volume

specifies a valid NonStop™ OS volume.

subvolume

specifies a valid NonStop™ OS subvolume.

Example

```
+ VOLUME $DATA.ASAP
```

Appendix A: Querying the ASAP Database

The ASAP database is a set of key-sequenced files that you can query using Enform or user-written programs. This section shows a sample standard Enform query and a sample ASAPCOL as an Enform server. The **ENFALL** file contains a complete set of sample queries.

Standard Enform Query

This example shows a standard Enform query:

```
?section cpu
!-----!
! Display cpu statistics by \node and cpu for each hour of the day.
! Note: this query must read all records in the database (no enform server)
!-----!
close net, net-srv, cpu-srv;

open cpu;

declare
x, cpubusy, cpuq, cpudisp, discraterate, chitraterate, swapraterate, memq, pcbmax, lcbmax;

set @target-records 0 @stats on;

title ""; subtitle "";

list by sysno noprint
  by cpuno noprint
  by ihour noprint
  x := ( (event.ts - 211024440000000000) /10000 * 65536 ) noprint
  cpubusy := avg (busy over ihour) noprint
  cpuq := avg (qlen over ihour) noprint cpudisp
:= avg (disps over ihour) noprint discraterate := avg
(discs over ihour) noprint chitraterate := avg (chits
over ihour) noprint swapraterate := avg (swaps over
ihour) noprint memq := avg (mqlen over ihour)
noprint pcbmax := max (pcb-use over ihour) noprint
lcbmax := max (lcb-use over ihour) noprint
after change on cpuno print skip 1 sysname space
```

```

"Cpu Date  Time  Busy Q  Disp Disc Chit Swap MQ Pcbx Lcbx"skip 1
"-----"

after change on ihour print

space 9

cpuno   as i3                               space
timestamp-date ( x ) as date "mb2/d2" space
timestamp-time ( x ) as time "h2:m2" space
cpubusy  as i4                               space
cpuq     as "[bz] i2"                         space
cpudisp  as "[bz] i4"                         space
discrate as "[bz] i4"                         space
chitrate as "[bz] i4"                         space
swaprata as "[bz] i4"                         space
memq     as "[bz] i2"                         space
pcbmax   as "[bz] i4"                         space
lcbmax   as "[bz] i4"                         space

;

```

ASAPCOL as an Enform Server

This example shows ASAPCOL as an Enform server:

```

?section cpu-last-hour
!-----!
! Display cpu statistics by \node and cpu for the most recent hour.
! Note: this query uses an enform server to quickly obtain stats.
! Run the stat server as follows: ASAPCOL/ NAME $CPUdb / DB <db prefix>
!-----!
close net, net-srv, cpu;
open cpu-srv;
declare
x, cpubusy, cpuq, cpudisp, discrate, chitrate, swaprata, memq, pcbmax, lcbmax;
set @target-records 0 @stats on;
title "System Cpu Date Time Busy Q Disp Disc Chit Swap MQ Pcbx Lcbx";
subtitle "-----";
list by ihour noprint
x := ( (event.ts - 211024440000000000) /10000 * 65536 ) noprint

```

```

cpubusy := avg (busy over ihour) noprint
cpuq := avg (qlen over ihour) noprint
cpudisp := avg (disps over ihour) noprint
discrate := avg (discs over ihour) noprint
chitrate := avg (chits over ihour) noprint
swaprate := avg (swaps over ihour) noprint
memq := avg (mqlen over ihour) noprint
pcbmax := max (pcb-use over ihour) noprint
lcbmax := max (lcb-use over ihour) noprint

```

after change on ihour print

```

sysname space
cpuno as i3 space
timestamp-date ( x ) as date "mb2/d2" space
timestamp-time ( x ) as time "h2:m2" space
cpubusy as i4 space
cpuq as "[bz] i2" space
cpudisp as "[bz] i4" space
discrate as "[bz] i4" space
chitrate as "[bz] i4" space
swaprate as "[bz] i4" space
memq as "[bz] i2" space
pcbmax as "[bz] i4" space
lcbmax as "[bz] i4" space

```

Appendix B: ASAP Data Definitions

This appendix contains the ASAP database data definitions and record layouts that are stored in a file named ASPDDLDB. These formats can change from one product version to the next.

DDL Section Definitions

Primary Key Definitions

The primary key is (*sysno* entity *its*), where *its* is a character-based inverted timestamp of the event. Entries are displayed from the most recent event to the oldest event, allowing efficient online access through the ASAP CI.

The ranking of the busiest processes is a part of the primary key for the process statistics entity. This attribute is redefined as the seventh byte of the *zdate* of the P^KEY.

```
DEF      P-key.                                !sys/cpu/netlh
03      Sysno      type binary 16.              !system number
03      Word       type binary 16 redefines sysno.
03      Line       type binary 16 occurs 4 times. !line name, or cpu
03      Cpuno      type binary 16 redefines Line. !cpu(<0 or >16=>all)
03      Linename   type character 8 redefines line. !line name for enform
03      Entity     type character 8 redefines line. !entity name (eg,DISK/FILE)
03      its        type binary 64.              !inverted timestamp
03      zdate      redefines its.              !inversion code rule:
04      zyear      pic x(1).                    !2200 - year
04      zmonth     pic x(1).                    ! 12 - month
04      zday       pic x(1).                    ! 31 - day
04      zhour      pic x(1).                    ! 24 - hour
04      zmin       pic x(1).                    ! 60 - min
04      zsec       pic x(1).                    ! 60 - sec
04      zrank      pic x(1).                    ! process busy rank
03      suffix     redefines its.              !
04      fillerx    type binary 16 occurs 3 times. ! cpu.p^key.suffix.
04      nssvnm     type binary 16.              !
END.
```

Event Definitions

```
DEF      Event.
02      Ts          type binary 64. !GMT usec start timestamp
02      Tsword     type binary 16 redefines Ts.
02      Word       type binary 16 redefines Ts.
02      Et         type binary 64. !elapsed time of event usecs
02      Etword     type binary 16 redefines Et.
END.
```

Query Key Definitions

```
DEF      Q-key.          !For Enform
02      ikey        pic x(18).
02      cpu-entity  redefines ikey.
03      csysno     type binary 16 display "m<\999.>".
03      ccpuno     type binary 16 display "m<99>".
02      lh-entity  redefines ikey.
03      lhsysno    type binary 16 display "m<\999.>".
03      lhlinename type character 8.
02      Entity     type character 8 redefines ikey.
02      iyear      pic x(11) redefines ikey.
02      imonth     pic x(12) redefines ikey.
02      iday       pic x(13) redefines ikey.
02      ihour      pic x(14) redefines ikey.
02      imin       pic x(15) redefines ikey.
02      isec       pic x(16) redefines ikey.
02      irank      pic x(17) redefines ikey.
END.
```

Extended Primary Key

The ASAP extended primary p-key with the domain name required for APP processing is:

```
DEF      P-key-x.
03      sysno      type binary 16.          !system number
03      word       type binary 16 redefines sysno. !word handle
03      domain     type binary 16 occurs 32 times. !domain name
03      dname      type character 64 redefines domain.
03      its        type binary 64.          !inverted timestamp
03      zdate      redefines its.          !inversion code
04      zyear      pic x(1).                !2200 - year
04      zmonth     pic x(1).                ! 12 - month
04      zday       pic x(1).                ! 31 - day
04      zhour      pic x(1).                ! 24 - hour
04      zmin       pic x(1).                ! 60 - min
04      zsec       pic x(1).                ! 60 - sec
04      zrank      pic x(1).                ! process busy rank
03      suffix     redefines its.          !
04      fillerx   type binary 16 occurs 3 times. ! cpu.p-key.suffix.
```



```

04      nssvnn  type binary 16.          !
END.

```

Extended Query Key

The ASAP extended queue q-key with the domain name required for APP processing is:

```

DEF      Q-key-x.                          !For Enform

02      ikey    pic x(74).
02      app-entity redefines ikey.
03      sysno   type binary 16 display "m<\999.>".
03      domain  type character 64.
02      entity  type character 66 redefines ikey.
02      iyear   pic x(67) redefines ikey.
02      imonth  pic x(68) redefines ikey.
02      iday    pic x(69) redefines ikey.
02      ihour   pic x(70) redefines ikey.
02      imin    pic x(71) redefines ikey.
02      isec    pic x(72) redefines ikey.
02      irank   pic x(73) redefines ikey.
END.

```

Application Entity Statistics Definitions

```

DEFINITION APP-STATS2.                    !Description.....Sect
02      system  type binary 16 occurs 4 times. !system name
02      word    type binary 16 redefines system. !word alignment
02      sysname type character 8 redefines system.
02      P-key-x type *.                       !sysno, domain & its
8.3
02      Q-key-x type * redefines P-key-x.     !enform q-key extended
02      event   type *.                       !ts, interval & info
02      Pid     type character 8.             !the process' name
02      Cpu     type binary 16.               !the process' cpu
02      Pin     type binary 16.               !the process' pin
02      Spare   type binary 16 occurs 5 times. !reserved
02      AggRec  type binary 16.               !aggregate rec indicator
02      Data-count type binary 16.           ! Number of Data occurrences
02      Op-Text type binary 16 occurs 8 times. !Op state text
02      OpText  type character 1 redefines op-text.
02      OpState type binary 16.               ! Operational state
02      Error   type binary 64.               ! Error (if any)
02      Error-state type binary 16.          ! Error state (or remove)
02      DitemCount type float 64.           ! Aggregate Count
02      Ditem   type float 64 occurs 12 times. ! Dataitems for
                                           Aggregation
02      Data    type sgp-metric-stats2 occurs 30 times.
END.

```

CIP Statistics Definitions

```

DEF ZClimData.

```

```

02 clim-present                type character 8.
02 word                        type binary 16 redefines clim-
present.
02 x-fabric-status            type character 8.
02 y-fabric-status            type character 8.
02 total-cipssrvs            type binary 16.
02 operation-mode            type character 8.
02 connection-points         type binary 32.
02 x2-fabric-status          type character 8.
02 y2-fabric-status          type character 8.
02 total-data-interface      type binary 32.
02 total-data-ipaddr         type binary 32.
02 total-mgmt-interface      type binary 32.
02 total-mgmt-ipaddr         type binary 32.
02 evt-log-entries           type binary 32.
02 restarts                  type binary 32.
02 total-processes           type binary 32.
02 agt-failures              type binary 32.
02 svr-failures              type binary 32.
02 agt-evt-log-entries       type binary 32.
02 agt-buffer-denials        type binary 64.
02 agt-itapi-errors          type binary 64.
02 agt-linux-errors          type binary 64.
02 agt-bfr-bytes-in-use-curr type binary 64.
02 agt-bfr-bytes-in-use-high type binary 64.
02 agt-total-recv-msgs       type binary 64.
02 agt-total-send-msgs       type binary 64.
02 agt-connected-cpus-curr   type binary 32.
02 agt-connected-cpus-high   type binary 32.
02 agt-queued-commands       type binary 32.
02 agt-failed-commands       type binary 32.
02 agt-last-itapi-errors     type binary 32.
02 agt-last-linux-errors     type binary 32.
02 srv-event-log-entries     type binary 32.
02 srv-buffer-denials        type binary 64.
02 srv-itapi-errors          type binary 64.
02 srv-linux-errors          type binary 64.
02 srv-bfr-bytes-in-use-curr type binary 64.
02 srv-bfr-bytes-in-use-high type binary 64.
02 srv-total-recv-msgs       type binary 64.
02 srv-total-send-msgs       type binary 64.
02 srv-total-bytes-sent       type binary 64.
02 srv-total-bytes-rcvd      type binary 64.
02 srv-connected-cpus-curr   type binary 32.
02 srv-connected-cpus-high   type binary 32.
02 srv-deferred-sends        type binary 32.
02 srv-failed-commands       type binary 32.
02 srv-last-itapi-errors     type binary 32.
02 srv-last-linux-errors     type binary 32.
02 srv-tcp-listen-skts-curr   type binary 32.
02 srv-tcp-listen-skts-high   type binary 32.
02 srv-udp-sockets-curr      type binary 32.
02 srv-udp-sockets-high      type binary 32.
02 srv-tcp-connections-curr   type binary 32.
02 srv-tcp-connections-high   type binary 32.
END.

```

```

DEF ZClimState.
    02 clim-present          type binary 16.
    02 word                  type binary 16 redefines clim-
present.
    02 x-fabric-status      type binary 16.
    02 y-fabric-status      type binary 16.
    02 total-cipssrvs      type binary 16.
    02 operation-mode      type binary 16.
    02 connection-points   type binary 16.
    02 x2-fabric-status     type binary 16.
    02 y2-fabric-status     type binary 16.
    02 total-data-iface    type binary 16.
    02 total-data-ipaddr   type binary 16.
    02 total-mgmt-iface    type binary 16.
    02 total-mgmt-ipaddr   type binary 16.
    02 evt-log-entries     type binary 16.
    02 restarts            type binary 16.
    02 total-processes     type binary 16.
    02 agt-failures        type binary 16.
    02 svr-failures        type binary 16.
    02 agt-evt-log-entries type binary 16.
    02 agt-buffer-denials  type binary 16.
    02 agt-itapi-errors    type binary 16.
    02 agt-linux-errors    type binary 16.
    02 agt-bfr-bytes-in-use-curr type binary 16.
    02 agt-bfr-bytes-in-use-high type binary 16.
    02 agt-total-recv-msgs type binary 16.
    02 agt-total-send-msgs type binary 16.
    02 agt-connected-cpus-curr type binary 16.
    02 agt-connected-cpus-high type binary 16.
    02 agt-queued-commands type binary 16.
    02 agt-failed-commands type binary 16.
    02 agt-last-itapi-errors type binary 16.
    02 agt-last-linux-errors type binary 16.
    02 srv-event-log-entries type binary 16.
    02 srv-buffer-denials  type binary 16.
    02 srv-itapi-errors    type binary 16.
    02 srv-linux-errors    type binary 16.
    02 srv-bfr-bytes-in-use-curr type binary 16.
    02 srv-bfr-bytes-in-use-high type binary 16.
    02 srv-total-recv-msgs type binary 16.
    02 srv-total-send-msgs type binary 16.
    02 srv-total-bytes-sent type binary 16.
    02 srv-total-bytes-rcvd type binary 16.
    02 srv-connected-cpus-curr type binary 16.
    02 srv-connected-cpus-high type binary 16.
    02 srv-deferred-sends  type binary 16.
    02 srv-failed-commands type binary 16.
    02 srv-last-itapi-errors type binary 16.
    02 srv-last-linux-errors type binary 16.
    02 srv-tcp-listen-skts-curr type binary 16.
    02 srv-tcp-listen-skts-high type binary 16.
    02 srv-udp-sockets-curr type binary 16.
    02 srv-udp-sockets-high type binary 16.
    02 srv-tcp-connections-curr type binary 16.
    02 srv-tcp-connections-high type binary 16.
END.

```

```

DEFINITION CLIM-Stats.
!Description.....Sect
  02 system          type binary 16  occurs 4 times.  ! system name
  02 word            type binary 16  redefines system.! word alignment handle
  02 sysname        type character 8 redefines system.
  02 P-key-x        type *.                ! sysno, domain & its
8.3
  02 Q-key-x        type * redefines P-key-x.        ! inform q-key extended
  02 event          type *.                ! ts, interval & info
  02 Type           type binary 16.         ! Record type CLIM, MON,
PROV
  02 AggRec         type binary 16.         ! AggRec indicator
  02 Spare          type binary 16  occurs 6 times.  ! reserved
  02 OpText         type binary 16  occurs 5 times.  ! Op state text
  02 Status         type character 1 redefines OpText.
  02 OpState        type binary 16.         ! Operational state
  02 Error          type binary 64.         ! Error (if any)
  02 ErrorState     type binary 16.         ! Error state (or
remove)
  02 Data           type ZCLIMData.         ! CIPCLIM data
  02 State          type ZCLIMState.       ! CIPCLIM states
END.

* CIP\MON\

```

```

02 size-32769-and-larger      type binary 32.
02 connected-clim-curr        type binary 32.
02 connected-clim-high        type binary 32.
02 rejected-connections       type binary 32.
02 deferred-sends             type binary 32.
02 total-messages-sent        type binary 64.
02 total-messages-recv       type binary 64.
END.

```

```

DEF ZMonState.
02 MemUsed                    type binary 16.
02 word                        type binary 16 redefines MemUsed.
02 primary-cpu                type binary 16.
02 primary-pin                type binary 16.
02 priority                    type binary 16.
02 qio-pool-current           type binary 16.
02 qio-pool-limit             type binary 16.
02 total-rcv-sckt-reqs        type binary 16.
02 total-rcv-sckt-reqs-errs   type binary 16.
02 total-send-sckt-reqs       type binary 16.
02 total-send-sckt-reqs-errs  type binary 16.
02 data-bytes-rcvd            type binary 16.
02 data-bytes-sent            type binary 16.
02 incoming-connections       type binary 16.
02 outgoing-connections       type binary 16.
02 tcp-listen-skts-curr        type binary 16.
02 tcp-listen-skts-high       type binary 16.
02 udp-sockets-curr           type binary 16.
02 udp-sockets-high           type binary 16.
02 tcp-connections-curr        type binary 16.
02 tcp-connections-high       type binary 16.
02 size-1-128                 type binary 16.
02 size-129-256               type binary 16.
02 size-257-512               type binary 16.
02 size-513-1024              type binary 16.
02 size-1025-2048             type binary 16.
02 size-2049-4096             type binary 16.
02 size-4097-8192             type binary 16.
02 size-8193-12288            type binary 16.
02 size-12289-16384           type binary 16.
02 size-16385-32768           type binary 16.
02 size-32769-and-larger      type binary 16.
02 connected-clim-curr        type binary 16.
02 connected-clim-high        type binary 16.
02 rejected-connections       type binary 16.
02 deferred-sends             type binary 16.
02 total-messages-sent        type binary 16.
02 total-messages-recv       type binary 16.
END.

```

```

DEFINITION MON-Stats.
!Description.....Sect
02 system                    type binary 16 occurs 4 times. ! system name
02 word                        type binary 16 redefines system.! word alignment handle
02 sysname                    type character 8 redefines system.
02 P-key-x                    type *. ! sysno, domain & its
8.3

```

```

02 Q-key-x          type * redefines P-key-x.          ! enform q-key extended
02 event            type *.                          ! ts, interval & info
02 Type             type binary 16.                  ! Record type CLIM, MON,
PROV
02 AggRec           type binary 16.                  ! AggRec indicator
02 Spare            type binary 16 occurs 6 times.   ! reserved
02 OpText           type binary 16 occurs 5 times.   ! Op state text
02 Status           type character 1 redefines OpText.
02 OpState          type binary 16.                  ! Operational state
02 Error            type binary 64.                  ! Error (if any)
02 ErrorState       type binary 16.                  ! Error state (or
remove)
02 Data             type ZMONData.                   ! CIPMON data
02 State            type ZMONState.                  ! CIPMON states
END.

* CIP\CLIM\<<clim-provider-name>
DEF ZPROVData.
02 clim             type character 24.
02 word             type binary 16 redefines clim.
02 prov-type        type character 8.
02 ip-family        type character 6.
02 tp-status        type character 8.
END.

DEF ZPROVState.
02 clim             type binary 16.
02 word             type binary 16 redefines clim.
02 prov-type        type binary 16.
02 ip-family        type binary 16.
02 tp-status        type binary 16.
END.

DEFINITION PROV-Stats.
!Description.....Sect
02 system           type binary 16 occurs 4 times.   ! system name
02 word             type binary 16 redefines system.! word alignment handle
02 sysname          type character 8 redefines system.
02 P-key-x          type *.                          ! sysno, domain & its
8.3
02 Q-key-x          type * redefines P-key-x.          ! enform q-key extended
02 event            type *.                          ! ts, interval & info
02 Type             type binary 16.                  ! Record type CLIM, MON,
PROV
02 AggRec           type binary 16.                  ! AggRec indicator
02 Spare            type binary 16 occurs 6 times.   ! reserved
02 OpText           type binary 16 occurs 5 times.   ! Op state text
02 Status           type character 1 redefines OpText.
02 OpState          type binary 16.                  ! Operational state
02 Error            type binary 64.                  ! Error (if any)
02 ErrorState       type binary 16.                  ! Error state (or
remove)
02 Data             type ZPROVData.                   ! CIPPROV data
02 State            type ZPROVState.                  ! CIPPROV states
END.

```

Comm Statistics Definitions

```

DEFINITION COM-STATS.                                !Description
02  system                                           type binary 16  occurs 4 times.  !system name
02  word                                              type binary 16  redefines system.!word alignment
02  sysname                                           type character 8 redefines system.
02  P-key                                             type *.        !sysno, domain & its
02  Q-key                                             type * redefines P-key. !enform q-key
02  event                                             type *.        !ts, interval & info
02  Invalid                                           type binary 16.  !<> 0 then reason invalid
02  PCpu                                              type binary 16.  !Backup cpu,-1 non nonstop
02  PCpu-State                                        type binary 16.  !BCPU state
02  PPin                                              type binary 16.  !Primary Cpu pin
02  BCpu                                              type binary 16.  !Backup cpu,-1 non nonstop
02  BCpu-State                                        type binary 16.  !BCPU state
02  BPin                                              type binary 16.  !Primary Cpu pin
02  LDev                                              type binary 32.  !Logical device# of line
02  Type                                              type binary 16.  !Device type.
02  SubType                                           type binary 16.  !Device Subtype.

02  Op-Text                                           type binary 16  occurs 5 times. !Operational Status
02  OpText                                           type character 1 redefines op-text.
02  OpState                                           type binary 16.  !Operational State

02  Line-Stat                                         type binary 16  occurs 4 times. !Line Status
02  LineStat                                         type character 1 redefines line-stat.
02  LineStat-State                                   type binary 16.  !LineStat state.

02  Prog-Name                                         type binary 16  occurs 13 times. ! Device program file
02  ProgName                                         type character 1 redefines Prog-Name.
02  P-Config                                          type binary 16  occurs 4 times. !Primary configuration
02  PConfig                                          type character 1 redefines P-Config.
02  B-Config                                          type binary 16  occurs 4 times. ! Backup configurati
02  BConfig                                          type character 1 redefines B-Config.

02  Requests                                          type binary 32.  ! Number of requests.
02  Requests-State                                   type binary 16.  ! Requests State.
02  BusyPct                                           type binary 16.  ! Read busy % * 10.
02  BusyPct-State                                    type binary 16.  ! ReadBsyPct State.
02  ReadBsyPct                                       type binary 16.  ! Read busy % * 10.
02  ReadBsy-State                                    type binary 16.  ! ReadBsyPct State.
02  WriteBsyPct                                       type binary 16.  ! Write busy % * 10.
02  WriteBsy-State                                   type binary 16.  ! WriteBsyPct State.
02  InBytes                                           type binary 64.  ! Number input bytes.
02  InBytes-State                                    type binary 16.  ! InBytes State.
02  OutBytes                                          type binary 64.  ! Number output bytes.
02  OutBytes-State                                   type binary 16.  ! OutBytes state.
02  InDataBytes                                       type binary 64.  ! Number data in bytes.
02  InDataBytes-State                                type binary 16.  ! InDataBytes State.
02  OutDataBytes                                       type binary 64.  ! Number data out bytes
02  OutDataBytes-State                               type binary 16.  ! OutDataBytes State.
02  Rate                                              type binary 64.  ! Total #Reads & Writes
02  Rate-State                                       type binary 16.  ! NumReads State.
02  NumReads                                          type binary 32.  ! Number of read.
02  NumReads-State                                   type binary 16.  ! NumReads State.
02  NumWrites                                          type binary 32.  ! Number of writes.
02  NumWrites-State                                  type binary 16.  ! NumWrites State.

```

```

02 NumRetries          type binary 32.      ! Number retries perfor
02 NumRetries-State   type binary 16.      ! NumRetries State.
02 NumTrans           type binary 32.      ! Number transasctions.
02 NumTrans-State     type binary 16.      ! NumTrans State.
02 AvgRespTime        type binary 64.      ! Avg Response time/xac
02 AvgRespTime-State  type binary 16.      ! AvgRespTime State.

02 Spare1             type binary 16.
02 Spare2             type binary 16.
02 Spare3             type binary 16.
02 Spare4             type binary 16.
END.

```

CPU Statistics Definitions

```

DEFINITION Cpu-stats.
02      System        type binary 16 occurs 4 times.!system name
02      Word          type binary 16  redefines system.
02      Sysname       type character 8 redefines system.
02      P-key         type *.          !sysno & cpuno & its
02      Q-key         type * redefines P-key.
02      Event         type *.          !timestamp & interval info
02      Cputype       type binary 16.  !cpu type 0=>nsi,1=>nsii,2=>txp
02      Invalid       type binary 16.  !if <> 0 then reason invalid
02      Busy          type binary 16.  !percent cpu busy
02      Data          type binary 16 unsigned redefines busy.
02      Qlen          type binary 16.  !cpu queue length
02      Swaps         type binary 16.  !cpu pagefault rate (per sec)
02      Mqlen        type binary 16.  !memory queue length
02      Disps         type binary 16.  !cpu dispatch rate
02      Send-busy     type binary 16 unsigned. ! % send busy(low byte)
02      ISBusy        redefines Send-busy.
03      IBusy         type binary 8 unsigned. !interrupt busy
03      SBusy         type binary 8 unsigned. !Send busy
02      Chits         type binary 16.  !cache hit rate
02      Discs         type binary 16.  !disc i/o rate
02      pcb-use       type binary 16.  !pcbs in use
02      pcb-cnf       type binary 16.  !pcbs configured
02      mem-use       type binary 32.  !pages locked
02      mem-cnf       type binary 32.  !pages configured
02      sys-use       type binary 16.  !syspool in use (D30+segs use)
02      sys-cnf       type binary 16.  !syspool configd (D30+max segs)
02      map-use       type binary 16.  !mappool in use
02      map-cnf       type binary 16.  !mappool configured
02      lcb-use       type binary 16.  !lcbs in use Cxx
02      pcbx-use      type binary 16 redefines lcb-use.!hgh pin use D30+
02      lcb-cnf       type binary 16.  !lcbs conf Cxx
02      pcbx-cnf      type binary 16 redefines lcb-cnf.!hgh pin cnf D30+
02      tle-use       type binary 16.  !tles in use
02      tle-cnf       type binary 16.  !tles configured
02      OpState       type binary 16.  !Cpu State
02      Busy-State    type binary 16.  !cpu busy state
02      Qlen-State    type binary 16.  !cpu queue length state
02      Swaps-State   type binary 16.  !cpu pagefault state
02      Mqlen-State   type binary 16.  !memory queue length state

```



```

02      Disps-State      type binary 16.  !cpu dispatch state
02      Chits-State      type binary 16.  !cache hit rate state
02      Discs-State      type binary 16.  !disc i/o rate state
02      pcb-use-State    type binary 16.  !pcbs in use state
02      mem-use-State    type binary 16.  !pages locked state
02      sys-use-State    type binary 16.  !syspool in use state
02      pcbx-use-State   type binary 16.  !High Pin pcbs in use state
02      tle-use-State    type binary 16.  !tles in use state
02      Blades.
03      UpCount          type character 1.
03      State            type character 1.
02      CpuName          type character 10.
02      brk-pnt          type binary 16.  !break points in use
02      brk-pnt-State    type binary 16.  !break points in use state
02      page-size        type binary 32.  !page size
02      mem-size         type binary 32.  !memory size
02      first-vrtl-page  type binary 32.  !first virtual page
02      swappable-pages  type binary 32.  !swappable pages
02      free-pages       type binary 32.  !free pages available
02      cur-locked-mem   type binary 32.  !locked memory
02      max-locked-mem   type binary 32.  !max lockable memory pgs
02      high-locked-mem  type binary 32.  !max virtual memory locked
02      mem-size-state   type binary 16.  !memory size state
02      first-vrtl-page-state type binary 16.  !1st virtual page state
02      swappable-pages-state type binary 16.  !swappable pages state
02      free-pages-state type binary 16.  !free pages state
02      cur-locked-mem-state type binary 16.  !cur locked mem state
02      max-locked-mem-state type binary 16.  !max locked mem state
02      high-locked-mem-state type binary 16.  !high locked mem state
02      IPU              type binary 16.  !Number of IPU's per Blade
02      IPUState         type binary 16.  !IPU State
02      spare1           type binary 16.  !Spare
02      spare2           type binary 16.  !Spare
02      spare3           type binary 16.  !Spare
02      spare4           type binary 16.  !Spare
02      spare5           type binary 16.  !Spare
02      spare6           type binary 16.  !Spare
END.

```

Disk Entity Statistics Definitions

```

DEFINITION DSK-STATS.
02      System          type binary 16 occurs 4 times. !sys name
02      Word            type binary 16  redefines system.
02      Sysname         type character 8 redefines system.
02      P-key          type *.          !Sysno & cpuno & its
02      Q-key          type * redefines P-key.
02      Event          type *.          !Timestamp/interval info
02      Cpu            type binary 16.  !Cpu number
02      Pin            type binary 16.  !Process number
02      Avail-space    type binary 32.  !Disk space available
02      Capacity       type binary 32.  !Total disk space
02      Ldev           type binary 16.  !Logical device#
02      Mirrored       type binary 16.  !1 if mirrored Volume
02      B-Splits       type binary 16.  !Block split tot 512-4096
02      Chit           type binary 16.  !Cache hits tot 512-4096

```

```

02      Cmiss          type binary 16. !Cache miss tot 512-4096
02      Qlen           type binary 16. !Request Queue
02      Qmax           type binary 16. !Request Qlen
02      Requests       type binary 32. !Total write, read, seek
02      Reqs-blckd     type binary 16. !Request blckd file locks
02      Swaps          type binary 16. !Swap rate
02      Volsm-Qlen     type binary 16. !Volume Semaphore queue
02      Volsm-Qmax     type binary 16. !Volume semaphore qlen
02      Extra          type binary 16 occurs 4 times.
02 Disk-devices      occurs 2 times.
03      Invalid        type binary 16. !If <> 0 reason invalid
03      In-Use         type binary 16. !Device being used
03      State          type character 1.!Up; Down; Revive; Etc
03      P-B            type character 2.!Path designtn p,b,etc
03      Ctrl           type binary 16. !Controller number
03      Unit           type binary 16. !Physical unit number
03      Reads          type binary 16. !Read rate
03      Writes         type binary 16. !Write rate
03      Seeks          type binary 16. !Seek rate
03      Rate           type binary 16. !Total write,read,seek
03      Busy           type binary 16. !Total write,read,seek
03      Write-busy     type binary 16. !Percent busy writing
03      Read-busy      type binary 16. !Percent busy reading
03      Seek-busy      type binary 16. !Percent busy seeking
03      Input-Kb       type binary 32. !Calc input bytes/sec
03      Output-Kb      type binary 32. !Calc output bytes/sec
03      Group          type binary 16. !Servernet Group number
03      Spare1         type binary 16.
03      Spare2         type binary 16.
03      Spare3         type binary 16.
03      Spare4         type binary 16.
02      OpState        type binary 16. !Logical volume State
02      Big-Frag       type binary 32. !Largest free space fragment.
02      Avail-Space-State type binary 16. !State of Available Space
02      B-Splits-State type binary 16. !Block spit state of 512-4096
02      Chit-State     type binary 16. !Cache hits State of 512-4096
02      Qlen-State     type binary 16. !Request Queue State
02      Requests-State type binary 16. !Requets State write,read,seek
02      Reqs-Blckd-State type binary 16. !Requet blckd State file locks
02      Swaps-State    type binary 16. !Swap State
02      Read-State     type binary 16. !Device read rate state
02      Write-State    type binary 16. !Device write rate state
02      Rate-State     type binary 16. !Device rate state
02      Busy-State     type binary 16. !Percent busy state
02      Write-Busy-State type binary 16. !Percent write busy state
02      Read-Busy-State type binary 16. !Percent busy reading state
02      Input-Kb-State type binary 16. !Input bytes/sec state
02      Output-Kb-State type binary 16. !Output bytes/sec state
02      Cpu-State      type binary 16. !State of disk primary cpu
02      P-Path-State   type binary 16. !State of disk primary path
02      M-Path-State   type binary 16. !State of disk mirror path
END.

```

Expand Entity Statistics Definitions

```

DEFINITION NET-stats.
02      System      type binary 16 occurs 4 times.!system name
02      Word        type binary 16  redefines system.
02      Sysname     type character 8 redefines system.
02      P-key       type *.          !sysno & $lname & its
02      Q-key       type * redefines P-key.
02      Event       type *.          !timestamp & interval info
02      Subtype     type binary 16. !0=>singleline,1=>multiline
02      Pid1        type binary 16. !primary (cpu,pin)
02      Pid2        type binary 16. !backup (cpu,pin)
02      Ldev        type binary 16. !logical device number
02      Invalid     type binary 16. !if<>0 then reason invalid
02      L2-Proto    type character 8. !Level 2 protocol
                                !-- LEVEL 2 -----
02      i-frames-sent type binary 32. !base + 0
02      Data        type binary 32 redefines i-frames-sent.
02      i-frames-rcvd type binary 32. !base + 1
02      s-frames-sent type binary 32. !base + 2
02      s-frames-rcvd type binary 32. !base + 3
02      u-frames-sent type binary 32. !base + 4
02      u-frames-rcvd type binary 32. !base + 5
                                !-- LEVEL 4 PKTS/FWD/LINKS
02      packets-sent type binary 32. !base + 6
02      packets-rcvd type binary 32. !base + 7
02      packets-fwd  type binary 32. !base + 8
02      fwd-packets  type binary 32. !base + 9
02      rlink-sent   type binary 32. !base + 10
02      rlink-rcvd   type binary 32. !base + 11
                                !-- LEVEL 4 Send Ops -----
                                !send-base=stat-base+12
02      conn-send    type binary 32. !send + 0
02      trace-send   type binary 32. !send + 1
02      ncpm-send    type binary 32. !send + 2
02      lrq-send     type binary 32. !send + 3
02      lcmp-send    type binary 32. !send + 4
02      can-send     type binary 32. !send + 5
02      ack-send     type binary 32. !send + 6
02      nak-send     type binary 32. !send + 7
02      enq-send     type binary 32. !send + 8
                                !-- LEVEL 4 Rcvd Ops -----
                                !rcvd-base = stat-base+21
02      conn-rcvd    type binary 32. !rcvd + 0
02      trace-rcvd   type binary 32. !rcvd + 1
02      ncpm-rcvd    type binary 32. !rcvd + 2
02      lrq-rcvd     type binary 32. !rcvd + 3
02      lcmp-rcvd    type binary 32. !rcvd + 4
02      can-rcvd     type binary 32. !rcvd + 5
02      ack-rcvd     type binary 32. !rcvd + 6
02      nak-rcvd     type binary 32. !rcvd + 7
02      enq-rcvd     type binary 32. !rcvd + 8
                                !-- LEVEL 1 BCC/NoFr/LineQ
02      bcc-err      type binary 32. !base + 30
02      no-frame-cnt type binary 32. !base + 31
02      line-quality  type binary 32. !base + 32

```

```

02      frame-cnt      type binary 32. !base + 33
02      error-cnt      type binary 32. !base + 34
                                !-- LEVEL 4 Buff/Msg use --
02      net-cur-io-buf type binary 32.!base + 35
02      net-max-io-buf type binary 32.!base + 36
02      net-lt64        type binary 32.!base + 37
02      net-lt128       type binary 32.!base + 38
02      net-lt256       type binary 32.!base + 39
02      net-lt512       type binary 32.!base + 40
02      net-lt1024      type binary 32.!base + 41
02      net-lt2048      type binary 32.!base + 42
02      net-lt4096      type binary 32.!base + 43
02      net-gt4096      type binary 32.!base + 44
02      net-cur-oos-buf type binary 32.!base+ 45
02      net-max-oos-buf type binary 32.!base+ 46
02      net-max-combined-buf type binary 32.!base+47
02      net-oos-timeouts type binary 32.!base+48
02      pchg-send       type binary 32.!base + 49
02      pchg-rcvd       type binary 32.!base + 50
02      net-pool-size   type binary 32.!base + 51
02      net-pool-fails type binary 32.!base + 52
02      time-factor     type binary 32.!base + 53 sbw d21
02      dat-bytes-sent  type binary 32.!base + 54
02      dat-bytes-rcvd type binary 32.!base + 55
02      quer-cmds-sent type binary 32.!base + 56
02      quer-cmds-rcvd type binary 32.!base + 57
02      quer-rsps-sent type binary 32.!base + 58
02      quer-rsps-rcvd type binary 32.!base + 59
02      frame-dropped  type binary 32.!base + 60
02      Mem-Low        type binary 32.!base + 61
02      Invalid-Frames type binary 32.!base + 62
02      next-sys       type binary 8.!base + 63.<0:7> sbw
02      modem-errs     type binary 8.!base + 63.<8:15> sbw
02      OpState        type binary 16.!base + 65
02      net-pool-state type binary 16.
02      net-pool-kb-state type binary 16.
02      net-pool-fails-state type binary 16.
02      no-frame-cnt-state type binary 16.
02      bcc-err-state  type binary 16.
02      nak-send-state type binary 16.
02      nak-rcvd-state type binary 16.
02      packets-sent-state type binary 16.
02      packets-rcvd-state type binary 16.
02      packets-fwd-state type binary 16.
02      fwd-packets-state type binary 16.
02      rlink-sent-state type binary 16.
02      rlink-rcvd-state type binary 16.
02      data-frames-sent-state type binary 16.
02      data-frames-rcvd-state type binary 16.
02      dat-bytes-sent-state type binary 16.
02      dat-bytes-rcvd-state type binary 16.
02      Conn-Cmds-sent-state type binary 16.
02      Conn-Cmds-rcvd-state type binary 16.
02      Conn-Rsps-sent-state type binary 16.
02      Conn-Rsps-rcvd-state type binary 16.
02      quer-cmds-sent-state type binary 16.
02      quer-cmds-rcvd-state type binary 16.

```

```

02      quer-rsps-sent-state type binary 16.
02      quer-rsps-rcvd-state type binary 16.
02      Mem-low-state      type binary 16.
02      Invalid-Frames-state type binary 16.
02      frame-dropped-state type binary 16.
02      Invalid-address-state type binary 16.
02      Client-OpState type binary 16.
02      Cpu-B              type binary 16.
02      Pin-B              type binary 16.
02      Cpu-State         type binary 16.
02      spare1            type binary 16. !Spare
02      spare2            type binary 32. !Spare
02      spare3            type binary 32. !Spare
02      spare4            type binary 32. !Spare
02      spare5            type binary 32. !Spare
02      spare6            type binary 32. !Spare
END.

```

File Entity Statistics Definitions

```

DEFINITION FIL-STATS.
                                !Description
02  system                      type binary 16  occurs 4 times.  !system name
02  word                        type binary 16  redefines system. !word alignment
02  sysname                     type character 8 redefines system.
02  P-key-x                     type *.
                                !sysno, domain & its
02  Q-key-x                     type * redefines P-key-x. !enform Q-key extended
02  event                       type *.
                                !ts, interval & info
02  Pid                         type character 8.
                                !Process' name
02  Cpu                         type binary 16.
                                !Process' cpu
02  Pin                         type binary 16.
                                !Process' pin
02  C-code                      type binary 16.
                                !Client, CI action flags
02  Op-Text                    type binary 16 occurs 8 times. !Operational Status
02  OpText                     type character 1 redefines op-text !Operational Status
02  OpState                    type binary 16.
                                !Operational State
02  Error                      type binary 64.
                                !Error condition, if any
02  Error-State                type binary 16.
                                !Error state, 0 for remove

02  Full                       type binary 16.
                                !Percent full
02  Full-State                 type binary 16.
                                !Full state

02  FEOF                       type binary 64.
                                !Eof in bytes
02  FEOF-State                 type binary 16.
                                !Eof state

02  Owner                      type binary 16.
                                !Left byte=group, right=user
02  Owner-State                type binary 16.
                                !Owner state

02  Security                   type binary 32.
                                !Security
02  Security-State             type binary 16.
                                !Security state

02  FOpen                      type binary 16.
                                !1=file is open
02  FOpen-State                type binary 16.
                                !Open state

02  TMFAudited                 type binary 16.
                                !1=audited
02  TMFAudited-State           type binary 16.
                                !TMFAudited state

02  Files                      type binary 32.
                                !Number of files in subvol

```

```

02 Files-State      type binary 16.          !Number of files state

02 FCode           type binary 16.          !File code
02 FFormat         type binary 16.          !1=Format1, 2=Format2
02 FType           type binary 16.          !Unstruct, relative, etc.
02 SQLType         type binary 16.          !Protection view, table, etc.
02 Partitions      type binary 16.          !Number of partitions
02 LastOpen        type binary 64.         !Last time file was opened LCT
02 Last Modif      type binary 64.         !Last time file was modified LCT

02 Spare1          type binary 16.
02 Spare2          type binary 16.
02 Spare3          type binary 16.
02 Spare4          type binary 16.
END.

```

Node Entity Statistics Definitions

DEFINITION Exp-End-to-End-Stats.

```

02 End-Sys-Num     type binary 8.
02 Num-of-Path     type binary 8.
02 Cur-Path-Hop-Count type binary 8.
02 Cur-Path-Neighbor-Num type binary 8.
02 Cur-Path-Ldev   type binary 16.
02 Link-sent       type binary 16.
02 Link-recv       type binary 16.
02 Pkts-sent       type binary 16.
02 Pkts-recv       type binary 16.
END.

```

DEFINITION NCP-STATS.

```

02 System          type binary 16 occurs 4 times.
02 Word            type binary redefines system.
02 Sysname         type character 8 redefines system.
02 P-key           type *.
02 Q-key           type * redefines P-key.
02 Event           type *.
02 Invalid         type binary 16.
                   !-- Expand End-to-End stats -
02 NumofNodes      type binary 16.  ! upto 256 end-to-end entries
02 End-Node-State type Exp-End-to-End-Stats Occurs 256 times.
END.

```

Process Statistics Definitions

DEFINITION Pro-Stats3. !Description.....Sect

```

02 system          type binary 16 occurs 4 times. ! system name
02 word            type binary 16 redefines system.! word alignment handle
02 sysname         type character 8 redefines system.
02 P-key-x         type *. ! sysno, domain & its 8.3
02 Q-key-x         type * redefines P-key-x. ! enform q-key extended
02 event           type *. ! ts, interval & info
02 Pid            type character 8. ! the process' name
02 Pin             type binary 16. ! Pin or 0
02 AggRec          type binary 16. ! AggRec indicator

```

```

02 Spare          type binary 16 occurs 6 times.    ! reserved
02 OpText         type binary 16 occurs 10 times.   ! Op state text
02 Status        type character 1 redefines OpText.
02 OpState       type binary 16.                  ! Operational state
02 Error         type binary 64.                  ! Error (if any)
02 ErrorState    type binary 16.                  ! Error state (or remove)
02 Cpu           type binary 16.                  ! pri cpu or 99
02 Cpustate      type binary 16.
02 BCpu         type binary 16.                  ! bkup cpu or 99
02 BCpustate     type binary 16.
02 BPin         type binary 16.                  ! bkup pin or 0
02 Pri          type binary 16.                  ! Priority
02 PriState     type binary 16.
02 Busy         type binary 64.                  ! Process busy 8.14.03
                                           implied decimal
02 BusyState    type binary 16.
02 CBusy       type binary 64.                  ! 6/18/08 IPU busy implied
decimal
02 CBusyState   type binary 16.
02 Affinity-w   type binary 16 occurs 4 times.    ! 6/18/08 IPU Affinity
02 Affinity     type character 1 redefines Affinity-w. ! 6/18/08 IPU
Affinity
02 AffState     type binary 16.
02 IPU         type binary 16.                  ! 6/18/08 IPU
02 IPUState    type binary 16.
02 BIPU       type binary 16.                  ! 6/18/08 Backup IPU
02 BIPUState   type binary 16.
02 P-State     type binary 16 occurs 4 times.    ! Process state
02 PState     type character 1 redefines P-State.
02 PStateState type binary 16.
02 W-State     type binary 16 occurs 4 times.    ! Wait state
02 WState     type character 1 redefines W-State.
02 WStateState type binary 16.
02 MPages     type binary 64.                  ! Mem pages
02 MPagesState type binary 16.
02 MSent      type binary 64.                  ! Msgs sent 8.14.03 implied
                                           decimal
02 MSentState type binary 16.
02 MRecv     type binary 64.                  ! Msgs received 8.14.03
                                           implied decimal
02 MRecvState type binary 16.
02 QLen      type binary 32.                  ! Queue length
02 QLenState type binary 16.
02 MQLen     type binary 32.                  ! Max Queue length
02 MQLenState type binary 16.
02 PFAULTS   type binary 64.                  ! Page Faults 8.14.03
                                           implied decimal
02 PFAULTSState type binary 16.
02 PF-Name    type binary 16 occurs 18 times.
02 PFName     type character 1 redefines PF-Name. !Program file name
02 PFNameLen  type binary 16.                  ! It's len, if non-zero use
                                           pfile name
02 CAID      type binary 16.                  ! CAID
02 PAID      type binary 16.                  ! PAID
02 PType     type binary 16.                  ! Process type
02 RC       type binary 16.                  ! Remote creator
02 CreateTS  type binary 64.                  ! Creation TS

```

```

02 SMode      type binary 16.           ! Stop mode
02 SReq       type binary 16.           ! Stop request mode
02 PFS        type binary 32.           ! Current PFS
END.

```

ProcessBusy Statistics Definitions

```

DEFINITION Pcb-stats.
02      System      type binary 16 occurs 4 times.!system name
02      Word        type binary 16 redefines system.
02      Sysname     type character 8 redefines system.
02      P-key       type *.              !sysno & cpuno & its
02      Q-key       type * redefines P-key.
02      Event       type *.              !timestamp & interval info
02      Cputype     type binary 16.      !0=>ns1,1=>ns2 etc
02      Invalid     type binary 16.      !if<>0 then reason invalid
                                           !-- process Stats--

02      Busy        type binary 16 unsigned.
02      pid         type binary 16 unsigned.
02      pri         type binary 16 unsigned.
02      flags       type binary 16 unsigned.
02      groupinfo   type binary 16 unsigned.
02      rfe         type binary 16 unsigned.
02      processname type character 6.
02      lcbs        type binary 16 unsigned.
02      Pages       type binary 16 unsigned.
02      Processtime type binary 64.
02      RcvQLen     type binary 16 unsigned.
02      MsgsRcvd    type binary 16 occurs 2 times.
02      Rmsgsgs     type binary 32 redefines MsgsRcvd.
02      MsgsSent    type binary 16 occurs 2 times.
02      Smsgsgs     type binary 32 redefines MsgsSent.
02      userinfo    type binary 16 unsigned.
02      rfe2        type binary 16 unsigned.
02      rfe3        type binary 16 unsigned.
02      rfe4        type binary 16 unsigned.
02      Elapsedtime type binary 64.
02      PgmFileName type binary 16 occurs 12 times.
02      ProgramName type character 24 redefines PgmFileName.
02      Hometerminal type binary 16 occurs 12 times.
02      HomeTerm    type character 24 redefines Hometerminal.
02      ErrorState  type binary 16 unsigned.
02      Busy-State  type binary 16 unsigned.
02      RcvQLen-State type binary 16 unsigned.
02      MsgsRcvd-State type binary 16 unsigned.
02      MsgsSent-State type binary 16 unsigned.
02      Pages-State type binary 16 unsigned.
02      Lcbs-State  type binary 16 unsigned.
02      CBusy       type binary 32.           !Avg of all IPU's Busy
02      CBusyState  type binary 16.           !State of CBusy
02      Affinity-V  type binary 16.
02      IPU         type binary 16.
02      Spare1     type Binary 16 unsigned.

```



```

02 Spare2 type Binary 16 unsigned.
02 Spare3 type Binary 16 unsigned.
02 Spare4 type Binary 16 unsigned.
02 Spare5 type Binary 16 unsigned.END.

```

RDF Statistics Definitions

```

DEFINITION RDF-STATS.
                                !Description
02 system type binary 16 occurs 4 times. !system name
02 word type binary 16 redefines system.!word alignment
02 sysname type character 8 redefines system.
02 P-key-x type *. !sysno, domain & its
02 Q-key-x type * redefines P-key-x. !enform q-key extended
02 event type *. !ts, interval & info
02 Pid type character 8. !Process' name
02 Pin type binary 16. !Process' pin
02 C-code type binary 16. !Client, CI action flags
02 Op-Status type binary 16 occurs 8 times. !Operational Status
02 OpStatus type character 1 redefines op-status. !Operational Status
02 OpState type binary 16. !Operational State
02 Error type binary 64. !Error code, if any
02 Error-State type binary 16. !Error state, 0 for remove
02 Priority type binary 16. !Process Priority
02 Priority-State type binary 16. !Priority State
02 PCpu type binary 16. !Primary CPU
02 PCpu-State type binary 16. !Primary CPU State
02 BCpu type binary 16. !Backup CPU
02 BCpu-State type binary 16. !Backup CPU State
02 Record-type type binary 16. !Type for this record
02 Primary-vol type binary 16 occurs 4 times. !Updater p-vol
02 Primary-volch type character 1 redefines Primary-vol.
02 Secondary-vol type binary 16 occurs 4 times. !Updater b-vol
02 Secondary-volch type character 1 redefines Secondary-vol.
02 Audittrail-index type binary 16. !Index of audittrail
02 Ai-volume type binary 16 occurs 12 times.!Audit Image file 02
Ai-volch type character 1 redefines Ai-volume.
02 Ai-sno type binary 32. !Sequence no
02 Ai-rba type binary 32. !RBA
02 Spare type binary 16 occurs 6 times. !reserved
02 RtdTime type binary 64. !Real Time Delay (RTD)
02 RtdTime-State type binary 16. !RTDTime State
02 RtdSecs-State type binary 16. !RTDSecs State
END.

```

Spooler Statistics Definitions

```

DEFINITION SPL-STATS.
                                !Description
02 system type binary 16 occurs 4 times. !system name
02 word type binary 16 redefines system.!word alignment
02 sysname type character 8 redefines system.
02 P-key-x type *. !sysno, domain & its
02 Q-key-x type * redefines P-key-x. !enform q-key extended
02 event type *. !ts, interval & info
02 Cpu type binary 16. !Process cpu, -1 if not running
02 Cpu-State type binary 16. !CPU state

```

```

02 BCpu                type binary 16.          !Backup cpu, -1 if not nonstop
02 BCpu-State          type binary 16.          !BCPU state
02 Priority             type binary 16.          !Executing priority of process.
02 Op-Text             type binary 16 occurs 12 times. !Operational Status
02 OpText             type character 1 redefines op-text. !Operational
                                Status
02 OpState            type binary 16.          !Operational State
02 Error              type binary 64.          !Error condition, if any
02 Error-State        type binary 16.          !Error state, 0 for remove
02 SplType            type binary 16.          !Indicates the type of
                                spooler obj.

02 NumObjects         type binary 16.          !Number of objects or subobjects
02 NumTotObjects      type binary 16.          !Total number of objects or subs

02 NumJobs            type binary 32.          !Number of jobs for component
02 NumJobs-State      type binary 16.          !NumJobs State
02 BatchJobs          type binary 32.          !Number of batch jobs for component
02 BatchJobs-State    type binary 16.          !BatchJobs State
02 OpenJobs           type binary 32.          !Number of jobs currently open
02 OpenJobs-State     type binary 16.          !OpenJobs State
02 PrintJobs          type binary 32.          !Number of jobs currently printing
02 PrintJobs-State    type binary 16.          !PrintJobs State
02 HoldJobs           type binary 32.          !Number of jobs currently on hold
02 HoldJobs-State     type binary 16.          !HoldJobs State

02 NumDevQ            type binary 32.          !Number of Device Queues used
02 NumDevQ-State      type binary 16.          !SplNumDevQ state
02 NumLocs            type binary 16.          !Number of locations defined
02 NumLocs-State      type binary 16.          !NumLocs State

02 NumDevs            type binary 16.          !Number of devices defined
02 SplNumCols         type binary 16.          !Number of collectors defined
02 SplNumPrts         type binary 16.          !Number of print
                                processes defined

02 NumFontJobs        type binary 16.          !Number of font jobs defined
02 NumFontJobs-State  type binary 16.          !NumFontJobs State

02 ColPctFull         type binary 16.          !%Full for collector
02 ColPctFull-State   type binary 16.          !ColPctFull State

02 Spare1             type binary 16.
02 Spare2             type binary 16.
02 Spare3             type binary 16.
02 Spare4             type binary 16.
END.

```

*--This extension layout is for the Spooler detail statistics:

```

DEFINITION SPL-EXT.
02 Buffer-int         type binary 16 occurs 25 times.
02 Supervisor-ext    redefines Buffer-int.
03 FName            type character 26.          !Supervisors Log File.
03 ColAllocSize     type binary 32.          !Collectors Allocated Size.
03 ColTotalSize     type binary 32.          !Collectors Total Avail Size.
02 Collector-ext     redefines Buffer-int.

```

```

03 FName          type character 26.          !Collectors File name.
03 UnitSize       type binary 16.
03 UnitAlloc      type binary 32.
03 TotalUnits     type binary 32.
02 Device-ext     redefines Buffer-int.
03 FName          type character 8.          !Associated Print Process name.
03 Flags          type binary 16.
03 FormName       type character 16.
03 RetryIntvl     type binary 16.
03 NumRetries     type binary 16.
03 RestartIntvl   type binary 16.
03 MaxRetries     type binary 16.
02 PrintProc-ext  redefines Buffer-int.
03 FName          type character 26.          !Print Process Program Name.
03 Flags          type binary 16.
END.

DEFINITION SPL-STATS-EXT.
02 SPL-STATS      type *.
02 SPL-EXT        type *.
END.

```

SWAP Statistics Definitions

```

DEFINITION Swp-Stats.                                !Description..Sect
02      System          type binary 16 occurs 4 times.!sys name
02      Word            type binary 16  redefines system.
02      Sysname         type character 8 redefines system.
02      P-key-x         type *.          !Sysno & cpuno & its
02      Q-key-x         type * redefines P-key-x.
02      Event           type *.          !Timestamp/interval info
02      Cpu             type binary 16.  !Cpu number
02      Invalid         type binary 16.
02      OpState         type binary 16.  !Swap State
02      num-files       type binary 16.  !high locked mem state
02      num-files-state type binary 16.  !high locked mem state
02      total-swap      type float 64.   !Total swap space
02      total-swap-state type binary 16.  !Total swap space state
02      total-reserved  type binary 32.   !Reserved cpu pages
02      total-rsvd-state type binary 16.  !Reserved cpu pages state
02      total-reservatns type binary 32.  !Reserved cpu pages
02      total-rsvp-state type binary 16.  !Reserved cpu pages state
02      total-aval-pages type binary 32.  !Available pages
02      total-aval-pages-state type binary 16.!Available pages state
02      pct-full        type binary 16.   !Percent of used pages
02      pct-full-state  type binary 16.   !Percent used pages state
02      pct-avail       type binary 16.   !Percent of free pages
02      pct-avail-state type binary 16.   !Percent free pages state
02      creates-ok      type binary 32.   !Creates ok
02      creates-ok-state type binary 16.  !Creates ok state
02      creates-fail    type binary 32.   !Creates failed
02      creates-fail-state type binary 16. !Creates failed state
02      release-ok      type binary 32.   !Releases ok
02      release-ok-state type binary 16.  !Releases ok state
02      release-fail    type binary 32.   !Releases failed
02      release-fail-state type binary 16. !Releases failed state
02      resize-ok       type binary 32.   !Resizes ok

```

```

02      resize-ok-state    type binary 16.      !Resizes ok state
02      resize-fail       type binary 32.      !Resizes failed
02      resize-fail-state type binary 16.      !Resizes failed state
02      alloc-ok          type binary 32.      !Allocates ok
02      alloc-ok-state    type binary 16.      !Allocates ok state
02      alloc-fail        type binary 32.      !Allocates failed
02      alloc-fail-state  type binary 16.      !Allocates failed state
02      frees-ok          type binary 32.      !Frees ok
02      frees-ok-state    type binary 16.      !Frees ok state
02      frees-fail        type binary 32.      !Frees failed
02      frees-fail-state  type binary 16.      !Frees failed state
02      Cpu-Swapfiles     occurs 19 times.
03      OpState          type binary 16.      !SwapFile State
03      Filename          type character 27. !Swapfile name
03      Filename-State    type Binary 16.      !Swapfile name state
03      name-len          type binary 16.      !Length of Swapfile name
03      pct-full          type binary 16.      !Percent of used pages
03      pct-full-state    type binary 16.      !Percent used pages state
03      pages             type binary 32.      !Total cpu pages
03      pages-state       type binary 16.      !Total cpu pages state
03      reserved          type binary 32.      !Reserved cpu pages
03      reserved-state    type binary 16.      !Reserved cpu pages state
03      available         type binary 32.      !Available pages
03      available-state   type binary 16.      !Available pages state
03      max-reserved      type binary 32.      !Max pages rsrvd in the f
03      max-reserved-state type binary 16.      !Max pages rsrvd state
03      Threshold         type binary 32.      !pages to generate msg 20
03      Threshold-state   type binary 16.      !threshold state
03      File-Error        type binary 16.      !Guardian File error number
02      SwapGBMB          type Character 2.    !Total swap in GB or MB
02      spare1            type binary 16.      !Spare
02      spare2            type binary 16.      !Spare
02      spare3            type binary 16.      !Spare
02      spare4            type binary 16.      !Spare
02      spare5            type binary 16.      !Spare
02      spare6            type binary 16.      !Spare
END.

```

Tape Statistics Definitions

```

DEFINITION TAP-STATS.                                     !Description
02  system          type binary 16  occurs 4 times.      !system name
02  word            type binary 16  redefines system.    !word alignment
02  sysname         type character 8 redefines system.
02  P-key           type *.          !sysno, domain & its
02  Q-key           type * redefines P-key.             !enform q-key
02  event           type *.          !ts, interval & info
02  Cpu             type binary 16.   !Process cpu, -1 if not running.
02  Cpu-State       type binary 16.   !CPU state
02  BCpu            type binary 16.   !Backup cpu, -1 if not nonstop.
02  BCpu-State      type binary 16.   !BCPU state
02  LDev            type binary 32.   !Logical device number of the drive.
02  SubType         type binary 16.   !Device Subtype. All
                                         tapes are type 4.
02  DriveStatus     type binary 16.   !Device status.

```

```

02 Op-Text          type binary 16 occurs 12 times.  !Operational status
02 OpText          type character 1 redefines op-text. !Operational
                                Status
02 OpState         type binary 16.                !Operational State

02 ACS             type binary 16.                !0 if not ACS type. -1 if it is.
02 NLCheck         type binary 16.                !-1 if NL check processing mode.
02 BLPCheck        type binary 16.                !-1 if BLP check processing mode.
02 MountReqs       type binary 16. !Number of outstanding tapemounts for
                                ! this drive.
02 MountReqs-State type binary 16.                !MountReqs state.
02 MountTime       type binary 16.                !#mins with mountreqs > 0.
02 MountTime-State type binary 16.                !MountTime state.
02 Mount-Tape      type binary 16 occurs 4 times.  ! name of tape on
                                device.
02 MountTape       type character 1 redefines Mount-Tape.
02 Tape-Status     type binary 16 occurs 4 times.  !status of tape
                                in drive.
02 TapeStatus      type character 1 redefines Tape-Status.
02 TapeStatus-State type binary 16.                !TapeStatus state.
02 Tape-Label      type binary 16 occurs 4 times.  !TapeLabel of
                                tape in drive.
02 TapeLabel       type character 1 redefines Tape-Label.
02 TapeLabel-State type binary 16.                !TapeLabel state.

02 Proc-Name       type binary 16 occurs 8 times.  ! Process name using
                                tape drive.
02 ProcName        type character 1 redefines Proc-Name.
02 Prog-Name       type binary 16 occurs 5 times.  ! Filename using
                                tape drive.
02 ProgName        type character 1 redefines Prog-Name.
02 UserId          type binary 16.                ! UserId using tape drive.

02 Spare1         type binary 16.
02 Spare2         type binary 16.
02 Spare3         type binary 16.
02 Spare4         type binary 16.
END.

```

*--This extension layout is for the Tape detail statistics.

```

DEFINITION TAP-EXT.
02 Density         type binary 16.                !Tapemount density.
02 Prot           type binary 16.                !Protected or not. Use SPI value.
02 InputOutput     type binary 16. !In, OUT of Extended. Use SPI value.
02 Action         type binary 16.                !Action for tapemount.
END.

```

TCP/IP Entity Statistics

```

DEF ZTCP6Data.    !tcpipv6
02 Sockets        type binary 32. ! Number of Sockets Open
02 Word           type binary 16 redefines Sockets.
02 SockSPI        type binary 32. ! Number of Sockets SPI
02 SockTCP        type binary 32. ! Number of Sockets TCP
02 SockUDP        type binary 32. ! Number of Sockets UDP
02 SockUnkn       type binary 32. ! Num Sockets New and Unsupported

```

```

02 SockLstn                type binary 32. ! Number of Sockets Listening
02 connattempt            type binary 32.
02 accepts                 type binary 32.
02 connects               type binary 32.
02 drops                  type binary 32.
02 conndrops              type binary 32.
02 closed                 type binary 32.
02 segstimed              type binary 32.
02 rttupdated              type binary 32.
02 delack                  type binary 32.
02 timeoutdrop            type binary 32.
02 rexmtimeo              type binary 32.
02 persisttimeo           type binary 32.
02 keeptimeo              type binary 32.
02 keepprobe              type binary 32.
02 keepdrops              type binary 32.
02 snd-total               type binary 32.
02 snd-pack                type binary 32.
02 snd-byte                type binary 32.
02 snd-rexmitpack         type binary 32.
02 snd-rexmitbyte         type binary 32.
02 snd-rexmitfast         type binary 32. !v6 only
02 snd-acks                type binary 32.
02 snd-probe               type binary 32.
02 snd-urg                 type binary 32.
02 snd-winup              type binary 32.
02 snd-ctrl                type binary 32.
02 bad-hdr-len            type binary 32. !v4 only
02 rcv-total               type binary 32.
02 rcv-pack                type binary 32.
02 rcv-byte                type binary 32.
02 rcv-badsum              type binary 32.
02 rcv-badoff              type binary 32.
02 rcv-short               type binary 32. !v6 only
02 badseqs                 type binary 32.
02 unack                   type binary 32.
02 timeout                 type binary 32.
02 noport                  type binary 32.
02 rcv-urg                 type binary 32.
02 rcv-dup-pack            type binary 32.
02 rcv-dup-byte            type binary 32.
02 rcv-part-dup-pack       type binary 32.
02 rcv-part-dup-byte       type binary 32.
02 rcv-out-pack            type binary 32.
02 rcv-out-byte            type binary 32.
02 rcv-after-win-pack      type binary 32.
02 rcv-after-win-byte      type binary 32.
02 rcv-after-close         type binary 32.
02 rcv-win-probe           type binary 32.
02 rcv-dup-ack             type binary 32.
02 rcv-ack-too-much        type binary 32.
02 rcv-ack-pack            type binary 32.
02 rcv-ack-byte            type binary 32.
02 rcv-win-update          type binary 32.
02 paws-drop               type binary 32.
02 predict-acks            type binary 32.
02 predict-data            type binary 32.

```

```

02 pcbcachemiss          type binary 32.
02 persistdrop           type binary 32.
02 badsyn                type binary 32.
02 droppedsyn            type binary 32.
02 sc-added              type binary 32.
02 sc-completed          type binary 32.
02 sc-timed-out          type binary 32.
02 sc-overflowed         type binary 32.
02 sc-reset              type binary 32.
02 sc-unreach            type binary 32.
02 sc-bucketoverflow     type binary 32.
02 sc-aborted            type binary 32.
02 sc-dupesyn            type binary 32.
02 sc-dropped            type binary 32.

```

END.

```

DEF ZTCP6State.    !tcpipv6
  02 Sockets          type binary 16. ! Number of Sockets Open
  02 Word             type binary 16 redefines Sockets.
  02 SockSPI          type binary 16. ! Number of Sockets SPI
  02 SockTCP          type binary 16. ! Number of Sockets TCP
  02 SockUDP          type binary 16. ! Number of Sockets UDP
  02 SockUnkn         type binary 16. ! Num Sockets New and Unsupported
  02 SockLstn        type binary 16. ! Number of Sockets Listening
  02 connattempt     type binary 16.
  02 accepts         type binary 16.
  02 connects        type binary 16.
  02 drops           type binary 16.
  02 conndrops       type binary 16.
  02 closed          type binary 16.
  02 segstimed       type binary 16.
  02 rttupdated      type binary 16.
  02 delack          type binary 16.
  02 timeoutdrop     type binary 16.
  02 rexmttimeo     type binary 16.
  02 persisttimeo   type binary 16.
  02 kepttimeo      type binary 16.
  02 keepprobe       type binary 16.
  02 keepdrops       type binary 16.
  02 snd-total       type binary 16.
  02 snd-pack        type binary 16.
  02 snd-byte        type binary 16.
  02 snd-rexmitpack  type binary 16.
  02 snd-rexmitbyte  type binary 16.
  02 snd-rexmitfast  type binary 16.
  02 snd-acks        type binary 16.
  02 snd-probe       type binary 16.
  02 snd-urg         type binary 16.
  02 snd-winup       type binary 16.
  02 snd-ctrl        type binary 16.
  02 bad-hdr-len     type binary 16. !v4 only
  02 rcv-total       type binary 16.
  02 rcv-pack        type binary 16.
  02 rcv-byte        type binary 16.
  02 rcv-badsum      type binary 16.
  02 rcv-badoff      type binary 16.
  02 rcv-short       type binary 16.

```

```

02 badseqs                type binary 16.
02 unack                  type binary 16.
02 timeout                type binary 16.
02 noport                 type binary 16.
02 rcv-urg                type binary 16.
02 rcv-dup-pack           type binary 16.
02 rcv-dup-byte           type binary 16.
02 rcv-part-dup-pack     type binary 16.
02 rcv-part-dup-byte     type binary 16.
02 rcv-out-pack          type binary 16.
02 rcv-out-byte          type binary 16.
02 rcv-after-win-pack    type binary 16.
02 rcv-after-win-byte    type binary 16.
02 rcv-after-close       type binary 16.
02 rcv-win-probe         type binary 16.
02 rcv-dup-ack           type binary 16.
02 rcv-ack-too-much      type binary 16.
02 rcv-ack-pack          type binary 16.
02 rcv-ack-byte          type binary 16.
02 rcv-win-update        type binary 16.
02 paws-drop             type binary 16.
02 predict-acks          type binary 16.
02 predict-data          type binary 16.
02 pcbcachemiss          type binary 16.
02 persistdrop           type binary 16.
02 badsyn                type binary 16.
02 droppedsyn            type binary 16.
02 sc-added              type binary 16.
02 sc-completed          type binary 16.
02 sc-timed-out          type binary 16.
02 sc-overflowed         type binary 16.
02 sc-reset              type binary 16.
02 sc-unreach            type binary 16.
02 sc-bucketoverflow     type binary 16.
02 sc-aborted            type binary 16.
02 sc-dupesyn            type binary 16.
02 sc-dropped            type binary 16.
END.

DEF ZUDP6Data.
02 in-pkts                type binary 32.
02 Word                   type binary 16 redefines in-pkts.
02 bad-hdr-len            type binary 32.
02 bad-xsum               type binary 32.
02 nosum                  type binary 32.
02 bad-len                type binary 32.
02 noport                 type binary 32.
02 noportbcast            type binary 32.
02 fullsock               type binary 32.
02 pcbcachemiss          type binary 32.
02 out-pkts               type binary 32.
02 in-pkts-dropped        type binary 32.
02 out-pkts-dropped      type binary 32.
END.

DEF ZUDP6State.
02 in-pkts                type binary 16.

```



```

02 Word                type binary 16 redefines in-pkts.
02 bad-hdr-len         type binary 16.
02 bad-xsum            type binary 16.
02 nosum               type binary 16.
02 bad-len             type binary 16.
02 noport              type binary 16.
02 noportbcst         type binary 16.
02 fullsock           type binary 16.
02 pcbbcachemiss      type binary 16.
02 out-pkts           type binary 16.
02 in-pkts-dropped    type binary 16.
02 out-pkts-dropped   type binary 16.
END.

```

```

DEF ZICMP6Data.
02 err                 type binary 32.
02 Word                type binary 16 redefines err.
02 old-short           type binary 32.
02 old-icmp            type binary 32.
02 OutEchoRply        type binary 32.
02 OutDstUnrch        type binary 32.
02 OutSrcQnch         type binary 32.
02 OutRedirect        type binary 32.
02 OutEcho            type binary 32.
02 OutTimeExcd        type binary 32.
02 OutParamPrb        type binary 32.
02 OutTimeStmp        type binary 32.
02 OutTimeSRpl        type binary 32.
02 OutInfoReq         type binary 32.
02 OutInfoRpl         type binary 32.
02 RouterSolicit      type binary 32.
02 badcode            type binary 32.
02 too-short          type binary 32.
02 xsum               type binary 32.
02 bad-len            type binary 32.
02 reflect            type binary 32.
02 InEchoRply         type binary 32.
02 InDstUnrch        type binary 32.
02 InSrcQnch          type binary 32.
02 InRedirect         type binary 32.
02 InEcho            type binary 32.
02 InTimeExcd        type binary 32.
02 InParamPrb        type binary 32.
02 InTimeStmp        type binary 32.
02 InTimeSRpl        type binary 32.
02 InInfoReq         type binary 32.
02 InInfoRpl         type binary 32.
02 RouterAdvertise    type binary 32.
02 bad-rap-subcode    type binary 32.
02 bad-addrlist       type binary 32.
02 bad-wpa            type binary 32.
02 rdisc-nrecorded    type binary 32.
END.

```

```

DEF ZICMP6State.
02 err                 type binary 16.
02 Word                type binary 16 redefines err.

```

```

02 old-short                type binary 16.
02 old-icmp                 type binary 16.
02 OutEchoRply              type binary 16.
02 OutDstUnrch              type binary 16.
02 OutSrcQnch               type binary 16.
02 OutRedirect              type binary 16.
02 OutEcho                  type binary 16.
02 OutTimeExcd              type binary 16.
02 OutParamPrb              type binary 16.
02 OutTimeStmp              type binary 16.
02 OutTimeSRpl              type binary 16.
02 OutInfoReq               type binary 16.
02 OutInfoRpl               type binary 16.
02 RouterSolicit            type binary 16.
02 badcode                  type binary 16.
02 too-short                type binary 16.
02 xsum                     type binary 16.
02 bad-len                  type binary 16.
02 reflect                  type binary 16.
02 InEchoRply               type binary 16.
02 InDstUnrch               type binary 16.
02 InSrcQnch                type binary 16.
02 InRedirect                type binary 16.
02 InEcho                   type binary 16.
02 InTimeExcd               type binary 16.
02 InParamPrb               type binary 16.
02 InTimeStmp               type binary 16.
02 InTimeSRpl               type binary 16.
02 InInfoReq                 type binary 16.
02 InInfoRpl                 type binary 16.
02 RouterAdvertise           type binary 16.
02 bad-rap-subcode           type binary 16.
02 bad-addrlist              type binary 16.
02 bad-wpa                   type binary 16.
02 rdisc-nrecorded          type binary 16.

```

END.

```

DEF ZIP6Data.
02 ztotal                   type binary 32.
02 Word      type binary 16 redefines ztotal.
02 bad-xsum                   type binary 32.
02 too-short                  type binary 32.
02 too-small                  type binary 32.
02 bad-hdr-len                type binary 32.
02 bad-len                    type binary 32.
02 frag                       type binary 32.
02 frag-drop                  type binary 32.
02 frag-timeout               type binary 32.
02 fwd                        type binary 32.
02 cant-fwd                   type binary 32.
02 redir-sent                 type binary 32.
02 out-pkts                   type binary 32.
02 noproto                    type binary 32.
02 delivered                  type binary 32.
02 localout                   type binary 32.
02 odropped                   type binary 32.
02 reassembled                type binary 32.

```

```

02 fragmented                type binary 32.
02 ofragmented              type binary 32.
02 cantfrag                 type binary 32.
02 badoptions               type binary 32.
02 noroute                  type binary 32.
02 badvers                  type binary 32.
02 rawout                   type binary 32.
02 badsrcintf               type binary 32.
02 fragoverflow             type binary 32.
END.

```

```

DEF ZIP6State.
02 ztotal                    type binary 16.
02 Word      type binary 16 redefines ztotal.
02 bad-xsum                type binary 16.
02 too-short               type binary 16.
02 too-small               type binary 16.
02 bad-hdr-len             type binary 16.
02 bad-len                 type binary 16.
02 frag                    type binary 16.
02 frag-drop               type binary 16.
02 frag-timeout            type binary 16.
02 fwd                     type binary 16.
02 cant-fw                 type binary 16.
02 redir-sent              type binary 16.
02 out-pkts                type binary 16.
02 noproto                 type binary 16.
02 delivered               type binary 16.
02 localout                type binary 16.
02 odropped                type binary 16.
02 reassembled             type binary 16.
02 fragmented              type binary 16.
02 ofragmented             type binary 16.
02 cantfrag                type binary 16.
02 badoptions               type binary 16.
02 noroute                  type binary 16.
02 badvers                  type binary 16.
02 rawout                   type binary 16.
02 badsrcintf              type binary 16.
02 fragoverflow             type binary 16.
END.

```

```

DEF ZRTE6Data.
02 bad-redir                type binary 32.
02 Word      type binary 16 redefines bad-redir.
02 dyn                      type binary 32.
02 new-gw                   type binary 32.
02 not-reach                type binary 32.
02 wild-card                 type binary 32.
END.

```

```

DEF ZRTE6State.
02 bad-redir                type binary 16.
02 Word      type binary 16 redefines bad-redir.
02 dyn                      type binary 16.
02 new-gw                   type binary 16.
02 not-reach                type binary 16.

```

```
02 wild-card                type binary 16.
END.
```

```
DEF ZQIO6Data.
02 data-mds                  type binary 32.
02 Word      type binary 16 redefines data-mds.
02 max-data-mds              type binary 32.
02 dup-mds                   type binary 32.
02 max-dup-mds               type binary 32.
02 no-data-mds               type binary 32.
02 no-dup-mds                type binary 32.
02 res-recov-run             type binary 32.
02 qio-limit-warns          type binary 32.
02 qio-driver-errs           type binary 32.
02 curr-pool-space           type binary 32.
02 max-pool-space            type binary 32.
02 pool-fails                type binary 32.
02 total-mbufs               type binary 32.
02 curr-mbufs-inuse          type binary 32.
02 max-mbufs-inuse           type binary 32.
02 mbuf-fails                type binary 32.
02 dup-driv-mds              type binary 32.
02 max-dup-driv-mds          type binary 32.
02 no-dup-driv-mds           type binary 32.
02 size-128                  type binary 32.
02 size-256                  type binary 32.
02 size-512                  type binary 32.
02 size-1024                 type binary 32.
02 size-2048                 type binary 32.
02 size-4096                 type binary 32.
02 size-8192                 type binary 32.
02 size-12288                type binary 32.
02 size-16384                type binary 32.
02 size-32768                type binary 32.
02 size-57344                type binary 32.
02 qio-so-ip                 type binary 16.
02 so-forced-close           type binary 32.
02 qio-so-que-fulls         type binary 32.
02 mdcopy-adj                type binary 32.
02 outhdr-adj                type binary 32.
02 outdata-adj               type binary 32.
END.
```

```
DEF ZQIO6State.
02 data-mds                  type binary 16.
02 Word      type binary 16 redefines data-mds.
02 max-data-mds              type binary 16.
02 dup-mds                   type binary 16.
02 max-dup-mds               type binary 16.
02 no-data-mds               type binary 16.
02 no-dup-mds                type binary 16.
02 res-recov-run             type binary 16.
02 qio-limit-warns          type binary 16.
02 qio-driver-errs           type binary 16.
02 curr-pool-space           type binary 16.
02 max-pool-space            type binary 16.
02 pool-fails                type binary 16.
```

```

02 total-mbufs                type binary 16.
02 curr-mbufs-inuse           type binary 16.
02 max-mbufs-inuse            type binary 16.
02 mbuf-fails                  type binary 16.
02 dup-driv-mds                type binary 16.
02 max-dup-driv-mds           type binary 16.
02 no-dup-driv-mds            type binary 16.
02 size-128                    type binary 16.
02 size-256                    type binary 16.
02 size-512                    type binary 16.
02 size-1024                   type binary 16.
02 size-2048                   type binary 16.
02 size-4096                   type binary 16.
02 size-8192                   type binary 16.
02 size-12288                  type binary 16.
02 size-16384                  type binary 16.
02 size-32768                  type binary 16.
02 size-57344                  type binary 16.
02 qio-so-ip                   type binary 16.
02 so-forced-close             type binary 16.
02 qio-so-que-fulls           type binary 16.
02 mdcopy-adj                  type binary 16.
02 outhdr-adj                  type binary 16.
02 outdata-adj                 type binary 16.

```

END.

```

DEF ZARP6Data.
02 snt-arpreq                  type binary 32.
02 Word      type binary 16 redefines snt-arpreq.
02 rcv-arpreq                  type binary 32.
02 snt-arprsp                  type binary 32.
02 rcv-arprsp                  type binary 32.
02 snt-invreq                  type binary 32.
02 rcv-invreq                  type binary 32.
02 snt-invrsp                  type binary 32.
02 rcv-invrsp                  type binary 32.
02 snt-naks                    type binary 32.
02 rcv-naks                    type binary 32.
02 snt-probes                  type binary 32.
02 rcv-probes                  type binary 32.

```

END.

```

DEF ZARP6State.
02 snt-arpreq                  type binary 16.
02 Word      type binary 16 redefines snt-arpreq.
02 rcv-arpreq                  type binary 16.
02 snt-arprsp                  type binary 16.
02 rcv-arprsp                  type binary 16.
02 snt-invreq                  type binary 16.
02 rcv-invreq                  type binary 16.
02 snt-invrsp                  type binary 16.
02 rcv-invrsp                  type binary 16.
02 snt-naks                    type binary 16.
02 rcv-naks                    type binary 16.
02 snt-probes                  type binary 16.
02 rcv-probes                  type binary 16.

```

END.

```

DEF ZIGMP6Data.
  02 rcv-total                type binary 32.
  02 Word    type binary 16 redefines rcv-total.
  02 rcv-tooshort            type binary 32.
  02 rcv-badsum              type binary 32.
  02 rcv-queries             type binary 32.
  02 rcv-badqueries         type binary 32.
  02 rcv-report              type binary 32.
  02 rcv-badreports         type binary 32.
  02 rcv-ourreports         type binary 32.
  02 snd-reports            type binary 32.
END.

```

```

DEF ZIGMP6State.
  02 rcv-total                type binary 16.
  02 Word    type binary 16 redefines rcv-total.
  02 rcv-tooshort            type binary 16.
  02 rcv-badsum              type binary 16.
  02 rcv-queries             type binary 16.
  02 rcv-badqueries         type binary 16.
  02 rcv-report              type binary 16.
  02 rcv-badreports         type binary 16.
  02 rcv-ourreports         type binary 16.
  02 snd-reports            type binary 16.
END.

```

```

DEF ZSOCK6Data.
  02 bcast-allcpus-sent     type binary 32.
  02 Word    type binary 16 redefines bcast-allcpus-sent.
  02 bcast-onecpu-sent      type binary 32.
  02 bcast-rcvd             type binary 32.
  02 migr-sent              type binary 32.
  02 migr-cont-sent         type binary 32.
  02 migr-rcvd              type binary 32.
  02 migr-cont-rcvd         type binary 32.
  02 migr-timedout          type binary 32.
  02 migr-uxp-rcvd         type binary 32.
  02 migr-cont-uxp-rcvd    type binary 32.
END.

```

```

DEF ZSOCK6State.
  02 bcast-allcpus-sent     type binary 16.
  02 Word    type binary 16 redefines bcast-allcpus-sent.
  02 bcast-onecpu-sent      type binary 16.
  02 bcast-rcvd             type binary 16.
  02 migr-sent              type binary 16.
  02 migr-cont-sent         type binary 16.
  02 migr-rcvd              type binary 16.
  02 migr-cont-rcvd         type binary 16.
  02 migr-timedout          type binary 16.
  02 migr-uxp-rcvd         type binary 16.
  02 migr-cont-uxp-rcvd    type binary 16.
END.

```

```

DEF ZMONGQ6Data.
  02 allmons-sent           type binary 32.

```

```

02 Word      type binary 16 redefines allmons-sent.
02 somemons-sent      type binary 32.
02 allbutme-sent      type binary 32.
02 dpas-rcvd         type binary 32.
02 migr-rcvd         type binary 32.
02 loopbk-rcvd       type binary 32.
02 ipfrag-rcvd       type binary 32.
02 unk-rcvd          type binary 32.
02 bad-frmt-rcvd     type binary 32.
END.

```

```

DEF ZMONGQ6State.
02 allmons-sent      type binary 16.
02 Word      type binary 16 redefines allmons-sent.
02 somemons-sent      type binary 16.
02 allbutme-sent      type binary 16.
02 dpas-rcvd         type binary 16.
02 migr-rcvd         type binary 16.
02 loopbk-rcvd       type binary 16.
02 ipfrag-rcvd       type binary 16.
02 unk-rcvd          type binary 16.
02 bad-frmt-rcvd     type binary 16.
END.

```

```

DEF ZLOOP6Data.
02 who-tcp-sent      type binary 32.
02 Word      type binary 16 redefines who-tcp-sent.
02 who-tcp-rcvd     type binary 32.
02 who-udp-sent     type binary 32.
02 who-udp-rcvd     type binary 32.
02 have-tcp-sent    type binary 32.
02 have-tcp-rcvd    type binary 32.
02 have-udp-sent    type binary 32.
02 have-udp-rcvd    type binary 32.
02 packet-sent      type binary 32.
02 packet-rcvd      type binary 32.
02 update-sent      type binary 32.
02 update-rcvd      type binary 32.
END.

```

```

DEF ZLOOP6State.
02 who-tcp-sent      type binary 16.
02 Word      type binary 16 redefines who-tcp-sent.
02 who-tcp-rcvd     type binary 16.
02 who-udp-sent     type binary 16.
02 who-udp-rcvd     type binary 16.
02 have-tcp-sent    type binary 16.
02 have-tcp-rcvd    type binary 16.
02 have-udp-sent    type binary 16.
02 have-udp-rcvd    type binary 16.
02 packet-sent      type binary 16.
02 packet-rcvd      type binary 16.
02 update-sent      type binary 16.
02 update-rcvd      type binary 16.
END.

```

```

DEF ZIP66Data.

```

```

02 rx-total                type binary 32.
02 Word                    type binary 16 redefines rx-total.
02 rx-idle-discards       type binary 32.
02 rx-toosmall            type binary 32.
02 rx-tooshort            type binary 32.
02 rx-badvers             type binary 32.
02 rx-errors              type binary 32.
02 rx-local               type binary 32.
02 rx-nonlocal            type binary 32.
02 rx-multicast           type binary 32.
02 rx-badalaligns        type binary 32.
02 rx-noproto             type binary 32.
02 rx-delivered           type binary 32.
02 rx-discarded           type binary 32.
02 rx-forwarded           type binary 32.
02 rx-tentative           type binary 32.
02 tx-local               type binary 32.
02 tx-fragmented          type binary 32.
02 tx-quota-exceeded      type binary 32.
02 tx-multicast           type binary 32.
02 tx-forward             type binary 32.
02 tx-attempts            type binary 32.
02 tx-failures            type binary 32.
02 tx-looped-mcasts       type binary 32.
02 trl-trfailure          type binary 32.
02 enc-allocfailed        type binary 32.
02 rip-noproto            type binary 32.
02 rip-nospace            type binary 32.
02 rip-delivered          type binary 32.
02 rip-allocfailed        type binary 32.
02 rip-tooshort           type binary 32.
02 rip-badcksum           type binary 32.
02 frag-packets           type binary 32.
02 frag-fragments        type binary 32.
02 frag-nofragpart        type binary 32.
02 frag-badhopbyhop       type binary 32.
02 frag-ufpbadalign      type binary 32.
02 frag-ufptoobig         type binary 32.
02 frag-hdrtoobig        type binary 32.
02 frag-jumbo             type binary 32.
02 frag-allocfailed       type binary 32.
02 reas-fragsrcvd         type binary 32.
02 reas-pktsreass         type binary 32.
02 reas-timeout           type binary 32.
02 reas-copy              type binary 32.
02 reas-ufptoobig         type binary 32.
02 reas-badfraglen        type binary 32.
02 reas-payloadtoobig     type binary 32.
02 reas-overlap           type binary 32.
02 reas-dupfrag           type binary 32.
02 reas-jumbo             type binary 32.
02 reas-allocfailed       type binary 32.
02 fwd-success            type binary 32.
02 fwd-baddst             type binary 32.
02 fwd-badsrc             type binary 32.
02 fwd-enobufs            type binary 32.
02 fwd-emsgsize           type binary 32.

```



```
02 fwd-eother                type binary 32.
END.
```

```
DEF ZIP66State.
02 rx-total                  type binary 16.
02 Word      type binary 16 redefines rx-total.
02 rx-idle-discards         type binary 16.
02 rx-toosmall              type binary 16.
02 rx-tooshort              type binary 16.
02 rx-badvers               type binary 16.
02 rx-errors                type binary 16.
02 rx-local                 type binary 16.
02 rx-nonlocal              type binary 16.
02 rx-multicast             type binary 16.
02 rx-badalaligns          type binary 16.
02 rx-noproto               type binary 16.
02 rx-delivered             type binary 16.
02 rx-discarded             type binary 16.
02 rx-forwarded             type binary 16.
02 rx-tentative             type binary 16.
02 tx-local                 type binary 16.
02 tx-fragmented           type binary 16.
02 tx-quota-exceeded        type binary 16.
02 tx-multicast             type binary 16.
02 tx-forward               type binary 16.
02 tx-attempts              type binary 16.
02 tx-failures              type binary 16.
02 tx-looped-mcasts        type binary 16.
02 trl-trfailure            type binary 16.
02 enc-allocfailed          type binary 16.
02 rip-noproto              type binary 16.
02 rip-nospace              type binary 16.
02 rip-delivered            type binary 16.
02 rip-allocfailed          type binary 16.
02 rip-tooshort             type binary 16.
02 rip-badcksum             type binary 16.
02 frag-packets             type binary 16.
02 frag-fragments           type binary 16.
02 frag-nofragpart          type binary 16.
02 frag-badhopbyhop         type binary 16.
02 frag-ufpbadalign         type binary 16.
02 frag-ufptoobig           type binary 16.
02 frag-hdrtoobig           type binary 16.
02 frag-jumbo                type binary 16.
02 frag-allocfailed         type binary 16.
02 reas-fragsrcvd           type binary 16.
02 reas-pktsreass           type binary 16.
02 reas-timeout             type binary 16.
02 reas-copy                 type binary 16.
02 reas-ufptoobig           type binary 16.
02 reas-badfraglen         type binary 16.
02 reas-payloadtoobig       type binary 16.
02 reas-overlap             type binary 16.
02 reas-dupfrag             type binary 16.
02 reas-jumbo                type binary 16.
02 reas-allocfailed         type binary 16.
02 fwd-success              type binary 16.
```

```

02 fwd-baddst          type binary 16.
02 fwd-badsrc          type binary 16.
02 fwd-enobufs         type binary 16.
02 fwd-emsgsize        type binary 16.
02 fwd-eother          type binary 16.
END.

```

```

DEF ZICMP66Data.
02 tx-error            type binary 32.
02 Word                type binary 16 redefines tx-error.
02 tx-error-drops     type binary 32.
02 tx-nobufs          type binary 32.
02 tx-rlimit-drops    type binary 32.
02 rx-total            type binary 32.
02 rx-info             type binary 32.
02 rx-error            type binary 32.
02 rx-tooshort        type binary 32.
02 rx-checksum         type binary 32.
02 rx-notifications   type binary 32.
02 rx-torawip6        type binary 32.
02 rcv-queries         type binary 32.
02 rcv-badqueries     type binary 32.
02 rcv-reports        type binary 32.
02 rcv-badreports     type binary 32.
02 rcv-ourreports     type binary 32.
02 in-dst-unreach     type binary 32.
02 in-packet-too-big  type binary 32.
02 in-time-exceeded   type binary 32.
02 in-param-prob      type binary 32.
02 in-echo-request    type binary 32.
02 in-echo-reply      type binary 32.
02 in-mld-query       type binary 32.
02 in-mld-report       type binary 32.
02 in-nd-router-advert type binary 32.
02 in-nd-neighbor-solicit type binary 32.
02 in-nd-neighbor-advert type binary 32.
02 in-nd-redirect     type binary 32.
02 out-dst-unreach    type binary 32.
02 out-packet-too-big type binary 32.
02 out-time-exceeded  type binary 32.
02 out-param-prob     type binary 32.
02 out-echo-request   type binary 32.
02 out-echo-reply     type binary 32.
02 out-mld-report     type binary 32.
02 out-mld-reduction  type binary 32.
02 out-nd-router-solicit type binary 32.
02 out-nd-neighbor-solicit type binary 32.
02 out-nd-neighbor-advert type binary 32.
02 nd-badsolicit      type binary 32.
02 nd-badadvertise    type binary 32.
02 nd-badoptions      type binary 32.
02 nd-badcachelookup  type binary 32.
02 nd-badcacheupdate  type binary 32.
02 nd-badredirect     type binary 32.
02 rtr-badsolicit     type binary 32.
02 rtr-badadvertise  type binary 32.
END.

```

```

DEF ZICMP66State.
  02 tx-error                type binary 16.
  02 Word                    type binary 16 redefines tx-error.
  02 tx-error-drops         type binary 16.
  02 tx-nobufs              type binary 16.
  02 tx-rlimit-drops        type binary 16.
  02 rx-total               type binary 16.
  02 rx-info                type binary 16.
  02 rx-error               type binary 16.
  02 rx-tooshort            type binary 16.
  02 rx-checksum            type binary 16.
  02 rx-notifications       type binary 16.
  02 rx-torawip6           type binary 16.
  02 rcv-queries            type binary 16.
  02 rcv-badqueries         type binary 16.
  02 rcv-reports            type binary 16.
  02 rcv-badreports         type binary 16.
  02 rcv-ourreports         type binary 16.
  02 in-dst-unreach         type binary 16.
  02 in-packet-too-big      type binary 16.
  02 in-time-exceeded       type binary 16.
  02 in-param-prob          type binary 16.
  02 in-echo-request        type binary 16.
  02 in-echo-reply          type binary 16.
  02 in-mld-query           type binary 16.
  02 in-mld-report          type binary 16.
  02 in-nd-router-advert    type binary 16.
  02 in-nd-neighbor-solicit type binary 16.
  02 in-nd-neighbor-advert  type binary 16.
  02 in-nd-redirect         type binary 16.
  02 out-dst-unreach        type binary 16.
  02 out-packet-too-big     type binary 16.
  02 out-time-exceeded      type binary 16.
  02 out-param-prob         type binary 16.
  02 out-echo-request       type binary 16.
  02 out-echo-reply         type binary 16.
  02 out-mld-report         type binary 16.
  02 out-mld-reduction      type binary 16.
  02 out-nd-router-solicit  type binary 16.
  02 out-nd-neighbor-solicit type binary 16.
  02 out-nd-neighbor-advert type binary 16.
  02 nd-badsolicit          type binary 16.
  02 nd-badadvertise        type binary 16.
  02 nd-badoptions          type binary 16.
  02 nd-badcachelookup     type binary 16.
  02 nd-badcacheupdate     type binary 16.
  02 nd-badredirect         type binary 16.
  02 rtr-badsolicit         type binary 16.
  02 rtr-badadvertise      type binary 16.
END.

```

```

DEFINITION TCP-Stats2.                                !Description.....Sect
  02 system      type binary 16  occurs 4 times.  ! system name
  02 word        type binary 16  redefines system.! word alignment handle
  02 sysname     type character 8 redefines system.
  02 P-key-x     type *.          ! sysno, domain & its

```

```

02 Q-key-x      type * redefines P-key-x.      ! inform q-key extended
02 event       type *.                      ! ts, interval & info
02 Type        type binary 16.             ! Record type
02 AggRec      type binary 16.             ! AggRec indicator
02 Spare       type binary 16 occurs 6 times. ! reserved
02 OpText      type binary 16 occurs 4 times. ! Op state text
02 Status      type character 1 redefines OpText.
02 OpState     type binary 16.             ! Operational state
02 Error       type binary 64.             ! Error (if any)
02 ErrorState  type binary 16.             ! Error state (or remove)
02 TCPData     type ZTCP6Data.
02 TCPState    type ZTCP6State.
02 UDPData     type ZUDP6Data.
02 UDPState    type ZUDP6State.
02 ICMPData    type ZICMP6Data.
02 ICMPState   type ZICMP6State.
02 IPData      type ZIP6Data.
02 IPState     type ZIP6State.
02 RTEData     type ZRTE6Data.
02 RTEState    type ZRTE6State.
02 QIOData     type ZQIO6Data.
02 QIOState    type ZQIO6State.
02 ARPData     type ZARP6Data.
02 ARPState    type ZARP6State.
02 IGMPData    type ZIGMP6Data.
02 IGMPState   type ZIGMP6State.
02 SOCKData    type ZSOCK6Data.
02 SOCKState   type ZSOCK6State.
02 MONGQData   type ZMONGQ6Data.
02 MONGQState  type ZMONGQ6State.
02 LOOPData    type ZLOOP6Data.
02 LOOPState   type ZLOOP6State.
02 IP6Data     type ZIP66Data.
02 IP6State    type ZIP66State.
02 ICMP6Data   type ZICMP66Data.
02 ICMP6State  type ZICMP66State.

```

END.

DEF zSubnet6Data.

```

02 has-alias   type character 4.
02 Word        type binary 16 redefines has-alias.
02 failover    type character 6.
02 sharedip    type character 6.
02 zbrother-name type character 8.
02 zmedia-state type character 4.
02 zsubnet-ipv6 type character 4.
02 zsubnet-ipv4 type character 4.
02 subnet-out-pkts type binary 32.
02 subnet-in-pkts type binary 32.
02 subnet-out-errs type binary 32.
02 subnet-in-errs type binary 32.
02 mmacerrors  type binary 32.
02 mmactimeouts type binary 32.
02 tcp-filters type binary 32.
02 tcp-filters-err type binary 32.
02 tcp-filters-dereg type binary 32.
02 udp-filters type binary 32.

```

```

02 udp-filters-err      type binary 32.
02 udp-filters-dereg   type binary 32.
02 port-filters-drop   type binary 32.
02 data-drop           type binary 32.
02 subnet-abort        type binary 32.
02 subnet-reset        type binary 32.
02 subnet-last-err     type binary 32.
02 media-lost          type binary 32.
END.

```

```

DEF zSubnet6State.
02 has-alias           type binary 16.
02 Word                type binary 16 redefines has-alias.
02 failover            type binary 16.
02 sharedip            type binary 16.
02 zbrother-name       type binary 16.
02 zmedia-state        type binary 16.
02 zsubnet-ipv6        type binary 16.
02 zsubnet-ipv4        type binary 16.
02 subnet-out-pkts    type binary 16.
02 subnet-in-pkts     type binary 16.
02 subnet-out-errs    type binary 16.
02 subnet-in-errs     type binary 16.
02 mmacerrors          type binary 16.
02 mmactimeouts        type binary 16.
02 tcp-filters         type binary 16.
02 tcp-filters-err    type binary 16.
02 tcp-filters-dereg  type binary 16.
02 udp-filters         type binary 16.
02 udp-filters-err    type binary 16.
02 udp-filters-dereg  type binary 16.
02 port-filters-drop   type binary 16.
02 data-drop           type binary 16.
02 subnet-abort        type binary 16.
02 subnet-reset        type binary 16.
02 subnet-last-err    type binary 16.
02 media-lost          type binary 16.
END.

```

```

DEFINITION SUBNET-Stats2.                                !Description.....Sect
02 system             type binary 16  occurs 4 times.  ! system name
02 word               type binary 16  redefines system.! word alignment handle
02 sysname            type character 8 redefines system.
02 P-key-x            type *.          ! sysno, domain & its
02 Q-key-x            type * redefines P-key-x.        ! inform q-key extended
02 event              type *.          ! ts, interval & info
02 Type               type binary 16.  ! Record type
02 AggRec             type binary 16.  ! AggRec indicator
02 Spare              type binary 16  occurs 6 times.  ! reserved
02 OpText             type binary 16  occurs 4 times.  ! Op state text
02 Status             type character 1 redefines OpText.
02 OpState            type binary 16.  ! Operational state
02 Error              type binary 64.  ! Error (if any)
02 ErrorState         type binary 16.  ! Error state (or remove)
02 Data               type ZSubnet6Data.
02 State              type ZSubnet6State.
END.

```

```

DEF ZROUTE6Data.
  02 zref-cnt      type binary 32.
  02 word         type binary 16 redefines zref-cnt.
  02 zsubnet-name type character 8.
  02 zallroute    type character 4.
  02 zsa-family   type character 6.
  02 zradix-type  type character 6.
  02 zusage       type binary 32.
END.

```

```

DEF ZROUTE6State.
  02 zref-cnt      type binary 16.
  02 word         type binary 16 redefines zref-cnt.
  02 zsubnet-name type binary 16.
  02 zallroute    type binary 16.
  02 zsa-family   type binary 16.
  02 zradix-type  type binary 16.
  02 zusage       type binary 16.
END.

```

```

DEFINITION ROUTE-Stats2.                                !Description.....Sect
  02 system       type binary 16  occurs 4 times.  ! system name
  02 word         type binary 16  redefines system.! word alignment handle
  02 sysname      type character 8 redefines system.
  02 P-key-x      type *.          ! sysno, domain & its
  02 Q-key-x      type * redefines P-key-x.        ! enform q-key extended
  02 event        type *.          ! ts, interval & info
  02 Type         type binary 16.  ! Record type
  02 AggRec       type binary 16.  ! AggRec indicator
  02 Spare        type binary 16  occurs 6 times.  ! reserved
  02 OpText       type binary 16  occurs 4 times.  ! Op state text
  02 Status       type character 1 redefines OpText.
  02 OpState      type binary 16.  ! Operational state
  02 Error        type binary 64.  ! Error (if any)
  02 ErrorState   type binary 16.  ! Error state (or remove)
  02 Data         type ZRoute6Data.
  02 State        type ZRoute6State.
END.

```

```

DEF ZPORT6Data.
  02 zprocname    type character 8.
  02 Word         type binary 16 redefines zprocname.
  02 Caid         type binary 16.
  02 Paid         type binary 16.
  02 objectfile   type character 34.
  02 zprim-cpu    type binary 16.
  02 zprim-pin    type binary 16.
  02 zprim-fnum   type binary 16.
  02 zbkup-cpu    type binary 16.
  02 zbkup-pin    type binary 16.
  02 zbkup-fnum   type binary 16.
  02 zproto       type character 6.
  02 zlport       type binary 16.
  02 zloc-addr    type binary 32.
  02 zfport       type binary 16.
  02 zforgn-addr  type binary 32.

```

```

02 zsend          type binary 16.
02 zrecv          type binary 16.
02 family        type binary 16.
02 zloc-addr6    type binary 64.
02 zloc-word     type binary 16 redefines zloc-addr6.
02 zloc-filler   type binary 64.
02 zforgn-addr6 type binary 64.
02 zforgn-word   type binary 16 redefines zforgn-addr6.
02 zforgn-filler type binary 64.
END.

```

```

DEF ZPORT6State.
02 zprocname     type binary 16.
02 Word          type binary 16 redefines zprocname.
02 Caid          type binary 16.
02 Paid          type binary 16.
02 objectfile    type binary 16. !dummy placeholder
02 zprim-cpu     type binary 16.
02 zprim-pin     type binary 16.
02 zprim-fnum    type binary 16.
02 zbkup-cpu     type binary 16.
02 zbkup-pin     type binary 16.
02 zbkup-fnum    type binary 16.
02 zproto        type binary 16.
02 zlport        type binary 16.
02 zloc-addr     type binary 16.
02 zfport        type binary 16.
02 zforgn-addr   type binary 16.
02 zsend         type binary 16.
02 zrecv         type binary 16.
02 family        type binary 16.
02 zloc-addr6    type binary 16.
02 zloc-filler   type binary 16.
02 zforgn-addr6 type binary 16.
02 zforgn-filler type binary 16.
END.

```

```

DEFINITION PORT-Stats2.                                !Description.....Sect
02 system        type binary 16 occurs 4 times.      ! system name
02 word          type binary 16 redefines system.    ! word alignment handle
02 sysname       type character 8 redefines system.
02 P-key-x       type *.                               ! sysno, domain & its
02 Q-key-x       type * redefines P-key-x.           ! enform q-key extended
02 event         type *.                               ! ts, interval & info
02 Type          type binary 16.                       ! Record type
02 AggRec        type binary 16.                       ! AggRec indicator
02 Spare         type binary 16 occurs 6 times.      ! reserved
02 OpText        type binary 16 occurs 4 times.      ! Op state text
02 Status        type character 1 redefines OpText.
02 OpState       type binary 16.                       ! Operational state
02 Error         type binary 64.                       ! Error (if any)
02 ErrorState    type binary 16.                       ! Error state (or remove)
02 Data          type ZPort6Data.
02 State         type ZPort6State.
END.

```

TCPIP TELNET Entity Description ASAP 2.8

* This defines the structure written to ASAPCOL by the TCPIP SGP

```
DEF ZTelnetData.
  02 zin-conn          type binary 32.
  02 Word              type binary 16 redefines zin-conn.
  02 ztotal-open      type binary 32.
  02 ztotal-net-data  type binary 32.
  02 zavg-actv-term   type binary 32.
  02 ztotal-spi-req   type binary 32.
  02 ztotal-user-data type binary 32.
  02 ztotal-net-req   type binary 32.
  02 ztotal-user-req  type binary 32.
END.
```

```
DEF ZTelnetState.
  02 zin-conn          type binary 16.
  02 Word              type binary 16 redefines zin-conn.
  02 ztotal-open      type binary 16.
  02 ztotal-net-data  type binary 16.
  02 zavg-actv-term   type binary 16.
  02 ztotal-spi-req   type binary 16.
  02 ztotal-user-data type binary 16.
  02 ztotal-net-req   type binary 16.
  02 ztotal-user-req  type binary 16.
END.
```

DEFINITION TELNET-Stats1.

```
!Description.....Sect
02 system          type binary 16 occurs 4 times. ! system name
  02 word          type binary 16 redefines system.! word alignment handle
  02 sysname       type character 8 redefines system.
  02 P-key-x       type *. ! sysno, domain & its 8.3
  02 Q-key-x       type * redefines P-key-x. ! enform q-key extended
  02 event         type *. ! ts, interval & info
  02 Type          type binary 16. ! Record type TCP,Subnet,Route,Telnet
  02 AggRec        type binary 16. ! AggRec indicator
  02 Spare         type binary 16 occurs 6 times. ! reserved
  02 OpText        type binary 16 occurs 4 times. ! Op state text
  02 Status        type character 1 redefines OpText.
  02 OpState       type binary 16. ! Operational state
  02 Error         type binary 64. ! Error (if any)
  02 ErrorState    type binary 16. ! Error state (or remove)
  02 Data          type ZTelnetData. ! TELNET data
  02 State         type ZTelnetState. ! TELNET states
END.
```

TCPIP PORT Entity Description ASAP 2.8

* This defines the structure written to ASAPCOL by the TCPIP SGP

```
DEF ZPortData.
  02 zprocname       type character 8.
  02 Word            type binary 16 redefines zprocname.
  02 Caid            type binary 16.
  02 Paid            type binary 16.
  02 objectfile      type character 34.
```



```

02 zprim-cpu          type binary 16.
02 zprim-pin         type binary 16.
02 zprim-fnum       type binary 16.
02 zbkup-cpu        type binary 16.
02 zbkup-pin        type binary 16.
02 zbkup-fnum       type binary 16.
02 zproto           type character 4.
02 zlport           type binary 16.
02 zloc-addr        type binary 32.
02 zfport           type binary 16.
02 zforgn-addr      type binary 32.
02 zsend            type binary 16.
02 zrecv            type binary 16.
END.

```

```

DEF ZPortState.
02 zprocname        type binary 16.
02 Word             type binary 16 redefines zprocname.
02 Caid             type binary 16.
02 Paid             type binary 16.
02 objectfile       type binary 16. !dummy placeholder
02 zprim-cpu        type binary 16.
02 zprim-pin        type binary 16.
02 zprim-fnum       type binary 16.
02 zbkup-cpu        type binary 16.
02 zbkup-pin        type binary 16.
02 zbkup-fnum       type binary 16.
02 zproto           type binary 16.
02 zlport           type binary 16.
02 zloc-addr        type binary 16.
02 zfport           type binary 16.
02 zforgn-addr      type binary 16.
02 zsend            type binary 16.
02 zrecv            type binary 16.
END.

```

```

DEFINITION PORT-Stats1.          !Description.....Sect
02 system           type binary 16 occurs 4 times. ! system name
02 word             type binary 16 redefines system.! word alignment handle
02 sysname          type character 8 redefines system.
02 P-key-x          type *.                ! sysno, domain & its 8.3
02 Q-key-x          type * redefines P-key-x. ! inform q-key extended
02 event            type *.                ! ts, interval & info
02 Type             type binary 16.        ! Record type TCP,Subnet,Route,Telnet
02 AggRec           type binary 16.        ! AggRec indicator
02 Spare            type binary 16 occurs 6 times. ! reserved
02 OpText           type binary 16 occurs 4 times. ! Op state text
02 Status           type character 1 redefines OpText.
02 OpState          type binary 16.        ! Operational state
02 Error            type binary 64.        ! Error (if any)
02 ErrorState       type binary 16.        ! Error state (or remove)
02 Data             type ZPortData.        ! PORT data
02 State            type ZPortState.       ! PORT states
END.

```

TELNET WINDOW Entity Description ASAP 2.8

* This defines the structure written to ASAPCOL by the TCPIP SGP

```
DEF ZWindowData.  
  02 zdata-transfer type binary 16.  
  02 Word           type binary 16 redefines zdata-transfer.  
  02 zuser-bytes   type binary 32.  
  02 znet-bytes    type binary 32.  
  02 zread-req     type binary 32.  
  02 zwrite-req    type binary 32.  
  02 zwrite-read-req type binary 32.  
  02 zctl-req      type binary 32.  
  02 zsetmode-req  type binary 32.  
  02 zcancel-req   type binary 32.  
  02 zloc-addr     type binary 32.  
  02 zforgn-addr   type binary 32.  
  02 zloc-port     type binary 16.  
  02 zforgn-port   type binary 16.  
  02 zservname     type character 8.  
  02 zwin-typ      type character 8.  
  02 znvloc-addr   type character 26.  
  02 znvrem-addr   type character 26.  
  02 zcaller-addr  type binary 32.  
  02 zloc-addr6    type binary 64.  
  02 zloc-word     type binary 16 redefines zloc-addr6.  
  02 zloc-filler   type binary 64.  
  02 zforgn-addr6  type binary 64.  
  02 zforgn-word   type binary 16 redefines zforgn-addr6.  
  02 zforgn-filler type binary 64.  
  02 zcaller-addr6 type binary 64.  
  02 zcaller-word  type binary 16 redefines zcaller-addr6.  
  02 zcaller-filler type binary 64.  
  02 Openers       type binary 32.  
  02 Processes     type binary 32.  
  02 Active        type binary 32.  
END.
```

```
DEF ZWindowState.  
  02 zdata-transfer type binary 16.  
  02 Word           type binary 16 redefines zdata-transfer.  
  02 zuser-bytes   type binary 16.  
  02 znet-bytes    type binary 16.  
  02 zread-req     type binary 16.  
  02 zwrite-req    type binary 16.  
  02 zwrite-read-req type binary 16.  
  02 zctl-req      type binary 16.  
  02 zsetmode-req  type binary 16.  
  02 zcancel-req   type binary 16.  
  02 zforgn-addr   type binary 16.  
  02 zloc-port     type binary 16.  
  02 zforgn-port   type binary 16.  
  02 zservname     type binary 16.  
  02 zwin-typ      type binary 16.  
  02 znvloc-addr   type binary 16.  
  02 zcaller-port  type binary 16.  
  02 zloc-addr6    type binary 16.  
  02 zloc-filler   type binary 16.  
  02 zforgn-addr6  type binary 16.
```

```

02 zforgn-filler    type binary 16.
02 zcaller-addr6   type binary 16.
02 zcaller-filler  type binary 16.
02 Openers         type binary 16.
02 Processes       type binary 16.
02 Active          type binary 16.
END.

```

```

DEFINITION WINDOW-Stats1.      !Description.....Sect
02 system           type binary 16 occurs 4 times. ! system name
02 word            type binary 16 redefines system.! word alignment handle
02 sysname         type character 8 redefines system.
02 P-key-x         type *.                ! sysno, domain & its 8.3
02 Q-key-x         type * redefines P-key-x.    ! enform q-key extended
02 event          type *.                ! ts, interval & info
02 Type           type binary 16.           ! Record type

```

```

TCP,Subnet,Route,Telnet
02 AggRec         type binary 16.           ! AggRec indicator
02 Spare         type binary 16 occurs 6 times. ! reserved
02 OpText        type binary 16 occurs 4 times. ! Op state text
02 Status        type character 1 redefines OpText.
02 OpState       type binary 16.           ! Operational state
02 Error         type binary 64.           ! Error (if any)
02 ErrorState    type binary 16.           ! Error state (or remove)
02 Data          type ZWindowData.         ! WINDOW data
02 State         type ZWindowState.        ! WINDOW states

```

END.

TELNET SERVICE Entity Description ASAP 2.8

* This defines the structure written to ASAPCOL by the TCPIP SGP

```

DEF ZServiceData.
02 ztotal-wins    type binary 32.
02 Word          type binary 16 redefines ztotal-wins.
02 ztotal-sessions type binary 32.
02 ztotal-aborts type binary 32.
02 ztotal-inuses type binary 32.

```

END.

```

DEF ZServiceState.
02 ztotal-wins    type binary 16.
02 Word          type binary 16 redefines ztotal-wins.
02 ztotal-sessions type binary 16.
02 ztotal-aborts type binary 16.
02 ztotal-inuses type binary 16.

```

END.

```

DEFINITION SERVICE-Stats1.
!Description.....Sect
02 system           type binary 16 occurs 4 times. ! system name
02 word            type binary 16 redefines system.! word alignment handle
02 sysname         type character 8 redefines system.
02 P-key-x         type *.                ! sysno, domain & its      8.3
02 Q-key-x         type * redefines P-key-x.    ! enform q-key extended
02 event          type *.                ! ts, interval & info
02 Type           type binary 16.           ! Record type TCP,Subnet,Route,Telnet
02 AggRec         type binary 16.           ! AggRec indicator
02 Spare         type binary 16 occurs 6 times. ! reserved

```

```

02 OpText      type binary 16 occurs 4 times.  ! Op state text
02 Status      type character 1 redefines OpText.
02 OpState     type binary 16.                ! Operational state
02 Error       type binary 64.                ! Error (if any)
02 ErrorState  type binary 16.                ! Error state (or remove)
02 Data        type ZServiceData.             ! SERVICE data
02 State       type ZServiceState.            ! SERVICE states
END.

```

TMF Entity Statistics Definitions

```

DEFINITION TMF-STATS.                                !Description
02 system      type binary 16 occurs 4 times.        !system name
02 word        type binary 16 redefines system.      !word alignment
02 sysname     type character 8 redefines system.
02 P-key-x     type *.                               !sysno, domain & its
02 Q-key-x     type * redefines P-key-x.             !enform Q-key extended
02 event       type *.                               !ts, interval & info
02 Pid         type character 8.                     !Process' name
02 Cpu         type binary 16.                       !Process' cpu
02 Pin         type binary 16.                       !Process' pin
02 C-code      type binary 16.                       !Client, CI action flags
02 Op-Text     type binary 16 occurs 8 times.        !Operational Status
02 OpText      type character 1 redefines op-text    !Operational Status
02 OpState     type binary 16.                       !Operational State
02 Error       type binary 64.                       !Error condition, if any
02 Error-State type binary 16.                       !Error state, 0 for remove

02 Tps         type float 64.                        !Transaction rate per second
02 Tps-State   type binary 16.                      !Tps state
02 BeginTrans  type binary 16.                      !Begin transaction status
02 BeginTrans-State type binary 16.                 !BeginTrans state
02 Catalog     type binary 16.                      !Catalog status

02 Trans       type binary 16.                      !Number of transactions
02 Trans-State type binary 16.                      !Owner state
02 Hung        type binary 16.                      !1=at least one trans is hung
02 Hung-State  type binary 16.                      !Hung state
02 Duration    type binary 32.                      !Transaction duration in seconds
02 Duration-State type binary 16.                   !Duration state

02 Type        type character 8.                    !MAT/AUXnn
02 Overflow    type binary 16.                      !1=overflow space in use
02 Overflow-State type binary 16.                   !Overflow state
02 AuditHold   type binary 16.                      !1=no more audit because no space
02 AuditHold-State type binary 16.                  !AuditHold state
02 Used        type binary 16.                      !Used percent of audit trail
02 Used-State  type binary 16.                      !Used state

02 Active      type binary 16.                      !1=audit dump active
02 Active-State type binary 16.                    !Active state

02 Spare1     type binary 16.
02 Spare2     type binary 16.
02 Spare3     type binary 16.

```

```
02 Spare4          type binary 16.
END.
```

DDL Section Record Declarations

ASAP Database Record Declarations

```
RECORD CPU.
  File is DBCPU. Def is cpu-stats. Key 0 is p-key.  End.
RECORD SYS.
  File is DBSYS. Def is cpu-stats. Key 0 is p-key.  End.
RECORD NET.
  File is DBEXP. Def is net-stats. Key 0 is p-key.  End.
RECORD PCB.
  File is DBPCB. Def is pcb-stats. Key 0 is p-key.  End.
RECORD NCP.
  File is DBNCP. Def is NCP-STATS. Key 0 is P-key.  End.
RECORD DSK.
  File is DBDSK. Def is dsk-stats. Key 0 is p-key.  End.
RECORD APP.
  File is DBAPP. Def is App-stats. Key 0 is p-key-x. End.
RECORD RDF.
  File is DBRDF. Def is rdf-stats. Key 0 is p-key-x. End.
RECORD FIL.
  File is DBFIL. Def is fil-stats. Key 0 is p-key-x. End.
RECORD SPL.
  File is DBSPL. Def is spl-stats. Key 0 is p-key-x. End.
RECORD TAP.
  File is DBTAP. Def is tap-stats. Key 0 is p-key.  End.
RECORD TMF.
  File is DBTMF. Def is tmf-stats. Key 0 is p-key-x. End.
RECORD PRO.
  File is DBPRO. Def is pro-stats. Key 0 is p-key.  End.
RECORD COM.
  File is DBCOM.  Def is Com-stats. Key 0 is p-key. End.
RECORD TCP.
  File is DBTCP.  Def is Tcp-stats2. Key 0 is p-key-x. End
RECORD SWP.
  File is DBSWP. Def is swp-stats. Key 0 is p-key-x.  End.
RECORD CIP.
  File is DBCIP. Def is clim-stats. Key 0 is p-key.  End.
```

Enform Server Declarations

Enform server declarations for fast hourly rollup:

```
RECORD CPU-SRV.
  File is $CPUDB. Def is cpu-stats. Sequence is Q-key.  End.
RECORD SYS-SRV.
  File is $$SYSDB. Def is cpu-stats. Sequence is Q-key.  End.
RECORD NET-SRV.
```

File is \$EXPDB. Def is net-stats. Sequence is Q-key. End.
RECORD PCB-SRV.
File is \$PCBDB. Def is pcb-stats. Sequence is Q-key. End.
RECORD NCP-SRV.
File is \$NCPDB. Def is NCP-STATS. Sequence is Q-Key. End.
RECORD DSK-SRV.
File is \$DSKDB. Def is dsk-stats. Sequence is Q-key. End.
RECORD APP-SRV.
File is \$APPDB. Def is app-stats. Sequence is Q-key-x. End.
RECORD RDF-SRV.
File is \$RDFDB. Def is rdf-stats. Sequence is Q-key-x. End.
RECORD FIL-SRV.
File is \$FILDB. Def is fil-stats. Sequence is Q-key-x. End.
RECORD SPL-SRV.
File is \$SPLDB. Def is spl-stats. Sequence is Q-key-x. End.
RECORD TAP-SRV.
File is \$TAPDB. Def is tap-stats. Sequence is Q-key. End.
RECORD TMF-SRV.
File is \$TMFDB. Def is tmf-stats. Sequence is Q-key-x. End.
RECORD PRO-SRV.
File is \$PRODB. Def is pro-stats. Sequence is Q-key. End.
RECORD COM-SRV.
File is \$COMDB. Def is Com-stats. Sequence is Q-key. End.
RECORD TCP-SRV.
File is \$TCPDB. Def is Tcp-stats2. Sequence is Q-key-x. End.
RECORD SWP-SRV.
File is \$SWPDB. Def is Swp-stats. Sequence is Q-key-x. End.
RECORD CIP-SRV.
File is \$CIPDB. Def is Clim-stats. Sequence is Q-key-x. End.

Appendix C: ASAP CI DOTs Warning Messages

The ASAP Command Interpreter will return warning messages in response to certain conditions it finds while processing Discrete Objective Thresholds (DOTs) settings, starting ASAP, or processing DOTs commands. This appendix describes those warning messages.

Actions enabled but Action Server is not started as SUPER.SUPER.

Cause

The Action server is enabled (SET ACTION ON) but ASAP is not running as SUPER.SUPER.

Effect

There might be actions that ASAP cannot perform. For example ASAP might not be able to issue SCF commands to abort objects.

Recovery

Make sure the ASAP user ID has authority to perform all needed actions.

The ASAP version on node is downrev and needs to be upgraded to fully support all the new features in this version of ASAP. Please enter HELP VERSIONS for more information on versioning.

Cause

The ASAP version on the specified remote node is older than the version on this node.

Effect

It is possible to run mixed versions of ASAP but the central collector node should always be the most current version. If it is not then ASAP might not be collecting data from the remote nodes.

Recovery

The HELP VERSIONS command provides more information on versioning issues.

The ASAP version on node is newer than this version. Please upgrade all systems to the most current ASAP version. Please enter HELP VERSIONS for more information on versioning.

Cause

The ASAP version on the specified remote node is newer than the version on this node.

Effect

It is possible to run mixed versions of ASAP but the central collector node should always be the most current version. If it is not then ASAP might not be collecting data from the remote nodes.

Recovery

The HELP VERSIONS command provides more information on versioning issues.

The Objectives database on node is downrev and must be converted to the latest format. Please enter HELP CONVERT for detailed instructions on converting the database.

Cause

The objectives database on the specified node was created by an earlier version of ASAP.

Effect

Newer versions of ASAP can probably read from an older Objectives database but they will not be able to write to down-rev databases. The older ASAP CI will be required to add or modify records in the old database.

Recovery

The database should be upgraded to the current version to fully support all the features in the current ASAP version. The HELP CONVERT command provides information on converting down-rev databases.

The Objectives database on node is newer than this ASAP version. Objectives databases need to be a consistent version. Please enter HELP VERSIONS and HELP CONVERT for more information.

Cause

The objectives database on the specified node was created by an newer version of ASAP.

Effect

The ASAP command interpreter on the current node will not be able to write to that database, and might not be able to read from the database if EDL changed in the newer version.

Recovery

The HELP VERSIONS and HELP CONVERT commands provides information on database versioning.

The current EDL version for the entity entity does not match the EDL found in the Objectives database on node. The Objective records need to be reloaded using the current EDL. Please enter HELP RELOAD for more information.

Cause

The ASAP Entity Definition Language (EDL) in use for the specified entity when one or more objectives was added to the database on the specified node does not match the current EDL definition for that entity.

Effect

The ASAP SGP for that entity might not be able to process goals for the entity and the ASAP command interpreter might not be able to display data for the entity.

Recovery

The objectives for that entity need to be deleted and re-added using the current EDL. The HELP RELOAD command provides information on how to reload objectives.

ASAP will auto-configure entity domains on node.

Cause

There are no monitored objects for the specified entity on the specified node.

Effect

ASAP will auto-configure domains for that entity, based on its auto-configuration rules.

Recovery

This might not be a problem. See MONITOR Usage Considerations for more information on auto-configuration.

ASAP will auto-configure all domains on node.

Cause

There are no monitored objects specified for any entity on the specified node.

Effect

All enabled entities on the node will auto-configure objects based on their auto-configuration rules.

Recovery

This might not be a problem. See MONITOR Usage Considerations for more information on auto-configuration.

ASAP will not auto-configure entity domains on node.

Cause

There is at least one monitored object specified for the entity so ASAP will not auto-configure domains for that entity.

Effect

ASAP will only monitor domains that you specify for the entity.

Recovery

This might not be a problem. See MONITOR Usage Considerations for more information on auto-configuration.

No entity domains will be monitored on node.

Cause

Monitored objects have been added to the ASAP database for the specified entity but they have all been turned off.

Effect

ASAP will not monitor any object for that entity on the node.

Recovery

This might not be a problem. See MONITOR Usage Considerations for more information on monitoring objects.

Use COMMIT to place database changes into production use.

Cause

A change has been made to the ASAP objectives database by adding, deleting or modifying a monitored object or goal.

Effect

The ASAP database has been updated but a running copy of ASAP has not loaded the changes.

Recovery

Enter the COMMIT command to place the database changes into production use.

Entity *entity* is not currently defined in EDL.

Cause

A entity defined in the ASAP database no longer exists in the current EDL.

Effect

ASAP cannot process objects or goals for the entity.

Recovery

Delete the old entity records. Add the entity definition back into the set of EDL definitions and then use "GOAL *entity* *, DELETE" (entered twice) to delete all the objective records. The entity definition can then be safely removed.

Domains must be added to the database before using On/Off

Cause

An attempt was made to turn monitoring off or on for an object before using the add option to actually add the monitored object to the database.

Effect

No operation is performed on the object.

Recovery

Add the object using the MONITOR command.

The current Objectives DB setting does not match the Objectives DB in use by the running copy of ASAP on node.

Current setting: filename1

Running copy: filename2

Cause

ASAP was started on the specified node using a different setting for the Objectives database filename (SET OBJECTIVESDB.)

Effect

The ASAP database in use by the running copy of ASAP is used instead of the current SET OBJECTIVESDB setting.

Recovery

Determine the location of the correct database and change the SET OBJECTIVESDB setting and/or restart ASAP to pick up the new setting.

Please set ObjectivesEventState to LOW, MEDIUM or HIGH when using escalating or de-escalating goals

Cause

An escalating or de-escalating goal was entered but the setting for SET OBJECTIVESEVENTSTATE is higher than the HIGH state.

Effect

EMS events and/or actions created by escalating or de-escalating goals might not be emitted.

Recovery

Change the setting for OBJECTIVESEVENTSTATE to LOW, MEDIUM or HIGH.

At least one attribute value (attribute-name) in an escalating or de-escalating goal does not escalate or de-escalate. Escalating goal values (<,<<,<<< operators) should increase and de-escalating goal values (>,>>,>>>) should decrease at each level to avoid unpredictable results.

Cause

An escalating or de-escalating goal was entered that specified incorrect values.

Effect

EMS events and/or actions can be created at unexpected times.

Recovery

For an escalating goal the values should increase at each level, and for a de-escalating goal the values should decrease at each level. Re-enter the goal command using values that properly increase or decrease depending on the operator type.