SYBASE®

Monitoring e-Biz Impact

e-Biz Impact

5.4.5

DOCUMENT ID: DC00306-01-0545-01

LAST REVISED: July 2005

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book

Audience

This book is for developers, system administrators, end users, new users, and existing users of e-Biz Impact.

How to use this book

This book contains these chapters:

- Chapter 1, “Overview,” provides an overview of e-Biz Impact monitoring functionality.
- Chapter 2, “Configuring the Global Console,” describes how to create and configure agents, and determine the objects to display in each view.
- Chapter 3, “Using the Global Console,” describes how to use Global Console views to monitor an e-Biz Impact implementation,. Chapter 4 also explains how to use command and control (CNC) requests from the Global Console to control and extract information about e-Biz Impact objects.
- Chapter 4, “Monitoring e-Biz Impact Events,” describes how to use the e-Biz Impact Event Monitor.
- Chapter 5, “Configuring and Using Alerts,” describes how to configure e-Biz Impact alerts and provides development information on using Open Transport-XML alerts.
- Appendix A, “Alert IDs,” identifies the types of alerts that can be generated from e-Biz Impact and lists their associated identification numbers.

Related documents

e-Biz Impact documentation The following documents are available on the Sybase™ Getting Started CD in the e-Biz Impact 5.4.5 product container:

- The e-Biz Impact installation guide explains how to install the e-Biz Impact software.
- The e-Biz Impact release bulletin contains last-minute information not documented elsewhere.

e-Biz Impact online documentation The following e-Biz Impact documents are available in PDF and DynaText format on the e-Biz Impact 5.4.5 SyBooks CD:

- The *e-Biz Impact Application Guide* provides information about the different types of applications you create and use in an e-Biz Impact implementation.
- The *e-Biz Impact Authorization Guide* explains how to configure e-Biz Impact security.
- *e-Biz Impact Command Line Tools* describes how to execute e-Biz Impact functionality from a command line.
- The *e-Biz Impact Configurator Guide* explains how to configure e-Biz Impact using the Configurator.
- The *e-Biz Impact Feature Guide* describes new features, documentation updates, and fixed bugs in this version of e-Biz Impact.
- The *e-Biz Impact Getting Started Guide* provides information to help you quickly become familiar with e-Biz Impact.
- The *Monitoring e-Biz Impact* (this book) explains how to use the Global Console, the Event Monitor, and alerts to monitor e-Biz Impact transactions and events. It also describes how e-Biz Impact uses the standard Simple Network Management Protocol (SNMP).
- *Java Support in e-Biz Impact* describes the Java support available in e-Biz Impact 5.4.5.
- The *e-Biz Impact MSG-IDE Guide* describes MSG-IDE terminology and explains basic concepts that are used to build Object Definition Language (ODL) applications.
- The *e-Biz Impact ODL Guide* provides a reference to Object Definition Language (ODL) functions and objects. ODL is a high-level programming language that lets the developer further customize programs created with the IDE tools.
- The *e-Biz Impact TRAN-IDE Guide* describes how to use the TRAN-IDE tool to build e-Biz Impact production objects, which define incoming data and the output transactions produced from that data.

Note The *e-Biz Impact ODL Application Guide* has been incorporated into the *e-Biz Impact ODL Guide*.

The *e-Biz Impact Alerts Guide*, the *e-Biz Impact SNMP Guide*, and the *e-Biz Impact Global Console Guide* have been combined into a new guide—*Monitoring e-Biz Impact*.

Adaptive Server Anywhere documentation The e-Biz Impact installation includes Adaptive Server® Anywhere, which is used to set up a Data Source Name (DSN) used with e-Biz Impact security and authorization. To reference Adaptive Server Anywhere documentation, go to the Sybase Product Manuals Web site at Product Manuals at http://www.sybase.com/support/manuals/, select SQL Anywhere Studio from the product drop-down list, and click Go.

Note the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

Other sources of information

Use the Sybase Getting Started CD, the SyBooks CD, and the Sybase Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

  Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

  Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

  To access the Sybase Product Manuals Web site, go to Product Manuals at http://www.sybase.com/support/manuals/.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ **Finding the latest information on product certifications**

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Select Products from the navigation bar on the left.
3. Select a product name from the product list and click Go.
4. Select the Certification Report filter, specify a time frame, and click Go.
5. Click a Certification Report title to display the report.

❖ **Creating a personalized view of the Sybase Web site (including support pages)**

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

❖ **Finding the latest information on EBFs and software maintenance**

1. Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.
2. Select EBFs/Maintenance. Enter user name and password information, if prompted (for existing Web accounts) or create a new account (a free service).
3. Select a product.
4. Specify a time frame and click Go.

5 Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Conventions

The syntax conventions used in this manual are:

Key	Definition
commands and methods	Command names, command option names, utility names, utility flags, Java methods/classes/packages, and other keywords are in lowercase Arial font.
variable	Italic font indicates: • Program variables, such as *myServer* • Parts of input text that must be substituted, for example: *Server*.log • File names
File | Save	Menu names and menu items are displayed in plain text. The vertical bar shows you how to navigate menu selections. For example, File | Save indicates "select Save from the File menu."
`package 1`	Monospace font indicates: • Information that you enter in a graphical user interface, at a command line, or as program text • Sample program fragments • Sample output fragments

Accessibility features

This document is available in an HTML version that is specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger.

For information about how Sybase supports accessibility, see Sybase Accessibility at http://www.sybase.com/accessibility. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

CHAPTER 1

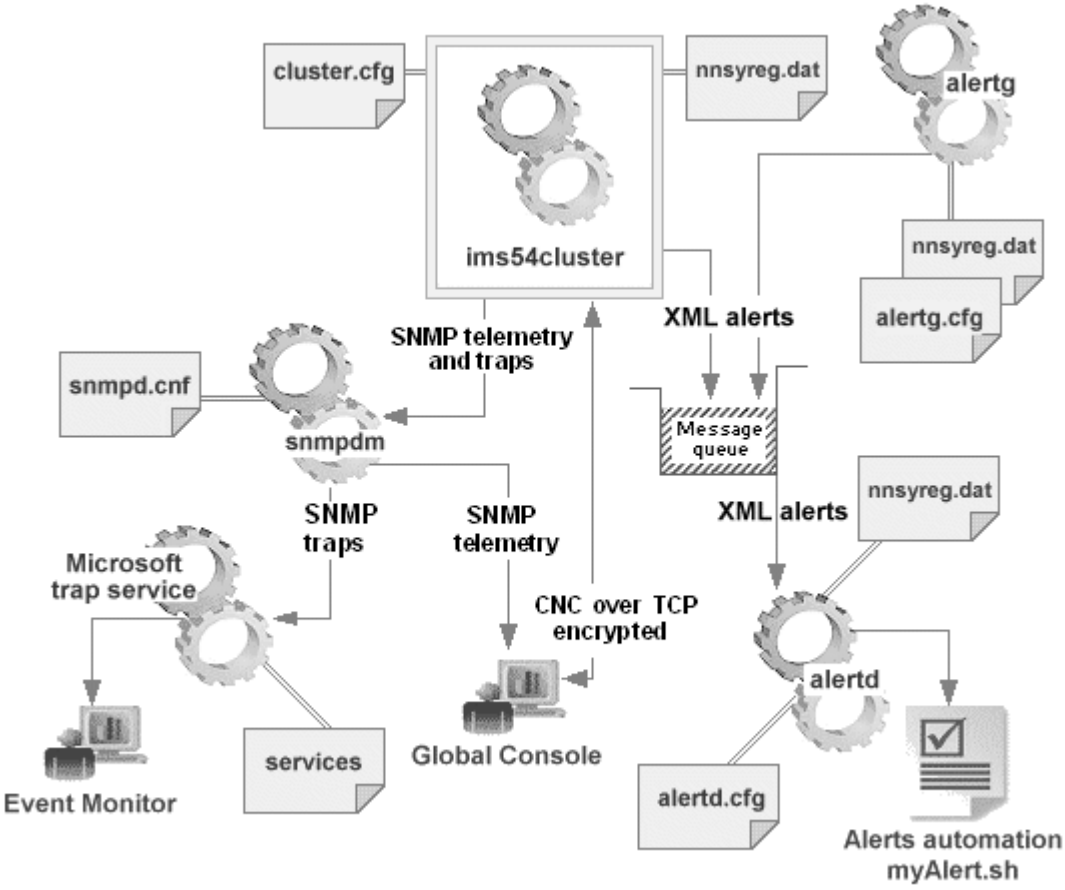
Overview

This chapter introduces the monitoring and alert capabilities available in e-Biz Impact. Subsequent chapters describe these capabilities in detail.

Introduction

e-Biz Impact lets you monitor cluster transactions and events in real time. System administrators can use this information to identify and resolve problems, and to tune for better performance. e-Biz Impact provides monitoring at several levels, as illustrated in Figure 1-1:

Figure 1-1: e-Biz Impact monitoring overview



e-Biz Impact monitoring includes:

- Global Console – displays Simple Network Management Protocol (SNMP) telemetry to a graphical user interface, which allows system administrators and operators to monitor all cluster objects on multiple e-Biz Impact servers.
- Event Monitor – displays SNMP traps (messages) collected from one or more e-Biz Impact servers to a graphical user interface. Events are triggered by e-Biz Impact and provide information, warnings, or errors that occur at runtime.

- Alerts – publishes SNMP traps and Open Transport XML alerts. An e-Biz Impact-provided file reads alert messages from an alert transport, such as a queue or file device, and invokes a user-maintained shell script or binary to act on the alert. Alerts provide a way for developers to respond programmatically to predefined cluster activities that require user intervention.

Note The monitoring options on the General tab of the Cluster Properties window in the Configurator affect what information is made available from each cluster to the SNMP daemon. See Chapter 2, "Configuring Clusters," in the *e-Biz Impact Configuration Guide*.

The Global Console and Event Monitor client user interfaces are available only on Windows, but both tools monitor e-Biz Impact servers on both Windows and UNIX systems.

Alerts are invoked from a command line or terminal window and are available on both Windows and UNIX systems.

Understanding SNMP

e-Biz Impact's alert functionality, Event Monitor, and Global Console all use the Simple Network Management Protocol (SNMP) to monitor and publish cluster activity, events, and alerts.

While the complete details of how SNMP works is outside the scope of this document, a basic understanding of SNMP as used with e-Biz Impact, is helpful.

SNMP basics

SNMP requires two basic elements to function: an agent and a manager. A Management Information Base (MIB) is also required, which provides a small set of commands for the exchange of information between the agent and the manager, and lists the unique object identifier (OID) of each managed element in an SNMP network.

- Agent – the devices (or software) that are managed. Provides the interface between the manager and the objects being managed or monitored. The agent typically stores and retrieves data as defined by the MIB and can asynchronously signal an event to the manager.

- Manager – provides the interface between the system operator and the management system. The manager typically queries agents, gets responses from agents, sets variables in agents, and acknowledges asynchronous events from agents.
- MIB – organized in a tree structure with individual variables, such as point status or description. A long numeric tag or object identifier (OID) is used to uniquely distinguish each variable in the MIB and in SNMP messages. The MIB provides five basic commands to communicate between the manager and the agent—Get, GetNext,GetResponse, Set, and Trap.

  The MIB serves as a data dictionary or codebook that is used to assemble and interpret SNMP messages.

Both the agent and manager implement the full SNMP communications protocols—the User Datagram Protocol (UDP) and the Internet Protocol (IP).

Note The UDP is the IP transport layer protocol that supports SNMP messages. Unlike TCP, UDP is a connectionless protocol. A UDP host places messages on the network without first establishing a connection with the recipient. UDP does not guarantee message delivery, but it is a lightweight protocol that can transport a large number of status messages without using too many network resources.

SNMP actions

When the SNMP manager and agent communicate:

1 The Get and GetNext messages allow the manager to request information for a specific variable.

2 The agent, upon receiving a Get or GetNext message, issues a GetResponse message to the manager with either the information requested or an error indication as to why the request cannot be processed.

3 A Set message allows the manager to request a change be made to the value of a specific variable in the case of an alarm remote that operates a relay.

4 The agent responds with a GetResponse message indicating the change has been made or an error indication as to why the change cannot be made.

5 The Trap message allows the agent to spontaneously inform the manager of an ìimportant event.

SNMP traps

Most of the messages (Get, GetNext, and Set) are issued only by the SNMP manager. Because the Trap message is the only command capable of being initiated by an agent, it is used by remote telemetry units (RTUs) to report alarms that notify the SNMP manager as soon as an alarm condition occurs instead of waiting for the SNMP manager to ask.

An SNMP trap is a change-of-state (COS) message issued by an SNMP agent that reports an event that can mean an alarm, a clear, or simply a status message.

e-Biz Impact SNMP implementation

To implement SNMP monitoring, e-Biz Impact uses:

- SNMP EMANATE Master Agent – the SNMP agent, which is installed with the e-Biz Impact server.
- Windows SNMP Trap Service – which you install separately. The SNMP Trap Service receives trap messages generated by local or remote SNMP agents and forwards the messages to SNMP management programs.

  Note When you install the Windows SNMP Trap Service, the Windows SNMP agent (SNMP Service) is also installed. e-Biz Impact uses the EMANATE Master Agent instead, so you disable the SNMP Service during monitoring configuration.

  The SNMP Trap Service daemon uses the *.cnf* files to translate the traps being received into a format the user can understand.
- Global Console and Event Monitor – the managers for e-Biz Impact SNMP monitoring, which receive and publish SNMP-generated messages.

Using the Global Console

To use the Global Console, you must perform the configuration steps in Chapter 2, “Configuring the Global Console.”

1 If you implement e-Biz Impact security, you must set up the Adaptive Server Anywhere authorization database on the e-Biz Impact server prior to configuring the Global Console. See the *e-Biz Impact Authorization Guide* for instructions.

2 Add the *ims* wrapper script location to the PATH environment variable. See “Setting the ims wrapper script path” on page 11.

3 Configure the directory for the port that the SNMP service or daemon uses. See “Configuring the SNMP port directory” on page 14.

4 Start the SNMP EMANATE Master Agent. See “Starting the SNMP EMANATE Master Agent” on page 14.

5 Configure e-Biz Impact clusters to publish telemetry to the SNMP service. See “Configuring clusters to publish SNMP telemetry” on page 19.

6 Start the cluster. See “Starting clusters” on page 20.

7 Start the Global Console. See “Starting the Global Console” on page 20.

8 Create an SNMP agent for each cluster you want to monitor. See “Creating SNMP agents” on page 20.

When you complete the configuration, you can begin monitoring cluster activity on e-Biz Impact servers. See Chapter 3, “Using the Global Console.”

Administrators or other authorized users can also issue command and control (CNC) requests from Global Console to restart clusters, reprocess problem transactions, and perform other cluster activities. See “Performing Global Console tasks” on page 49.

Using the Event Monitor

To use the Event Monitor, you must perform the configuration steps in “Setting up event monitoring,” in Chapter 4, “Monitoring e-Biz Impact Events.”

1 Install the Microsoft SNMP Trap Service. See “Installing the Microsoft SNMP Trap Service” on page 63.

2 Disable the SNMP Service. See “Disabling the Windows SNMP Service” on page 64.

3 Configure the SNMP EMANATE Master Agent service. This step is necessary only if you are sending traps to a custom port or multiple machines. See “Configuring the SNMP EMANATE Master Agent service” on page 64.

4 Configure e-Biz Impact clusters to publish alerts to the SNMP service. See “Configuring clusters to publish SNMP alert traps” on page 70.

When you complete the configuration steps, you can view informational, warning, and error conditions gathered from SNMP traps. See “Monitoring events” on page 71.

Using alerts

Support for the e-Biz Impact Open Transport-XML alert consists three applications:

- ims54alertg (UNIX, Windows)
- ims54alertd (UNIX, Windows)
- ims54alertdsvc (Windows only)

To simulate and generate OT-XML alerts, e-Biz Impact uses the ims54cluster binary and the ims54alertg binary.

Both ims54cluster and ims54alertg can publish OT-XML alerts. After an alert is published, an external handler is required to manage the alert, and take some user-defined action such as issuing e-mail or logging the event. The external handler used depends on the type of alert published.

To use the alert information from ims54cluster or ims54alertg, an OT-XML message on the alert transport, such as an item in the PENDING queue, must be turned into a user-defined action using the utility program.

ims54alertd reads alert messages from the alert transport, such as a queue or file device, and invokes a user-maintained shell script or binary. Use ims54alertdsvc to run as a Windows service. On UNIX systems, run ims54alertd either in the foreground or in the background, for example, as a daemon.

Before invoking the user script or binary, ims54alertd places the relevant information from the alert message in environment variables. While the user script or binary is executing, you can query the environment variables to control the execution flow in a script or binary.

After successfully processing the alert message from the transport device using the script or binary, ims54alertd can, optionally, copy the alert message to another transport device, such as a log device, for auditing or history purposes. Alert messages that cannot be processed are moved to an error transport device.

To configure and implement alert notification and response, see Chapter 5, “Configuring and Using Alerts.”

To view possible alert messages, see Appendix A, “Alert IDs.”

CHAPTER 2

Configuring the Global Console

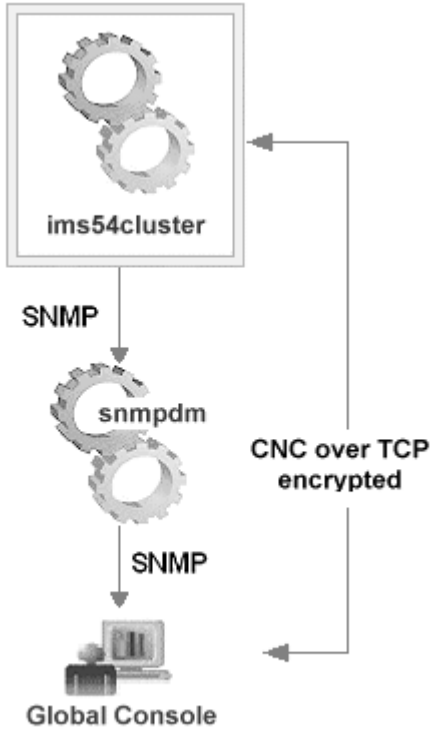
This chapter explains how to set up SNMP monitoring agents in the Global Console and define the objects that display in monitoring views.

Note Other SNMP management consoles, such as HP OpenView, IBM Tivoli, and Sun Net Manager can be used in place of the Global Console.

Introduction

The Global Console allows you to monitor multiple e-Biz Impact servers using the Simple Network Management Protocol (SNMP).

Figure 2-1: Global Console process flow



Use the Global Console to:

- Add, remove, or rename SNMP monitoring agents.
- Connect agents to and disconnect agents from e-Biz Impact servers to monitor the status of running clusters.
- Execute administrative control and command (CNC) requests against various server components (clusters, controllers, and applications).
- Control which roles are allowed to perform different monitoring tasks if you use e-Biz Impact optional authorization capabilities. See the *e-Biz Impact Authorization Guide* for instructions.

Note e-Biz Impact monitors e-Biz Impact server clusters using the Microsoft SNMP Trap Service and publishes the information to the Global Console using the SNMP EMANATE Master Agent service.

Configuring the Global Console

Before you use the Global Console to monitor your e-Biz Impact cluster activity, you must perform these configuration steps:

1 If you implement e-Biz Impact security, you must set up the Adaptive Server Anywhere authorization database on the e-Biz Impact server prior to configuring the Global Console. See the *e-Biz Impact Authorization Guide* for instructions.

2 Add the *ims* wrapper script location to the path where the cluster will run. See "Setting the ims wrapper script path" on page 11.

3 Configure the directory for the port that the SNMP service or daemon uses. See "Configuring the SNMP port directory" on page 14.

4 Start the SNMP EMANATE Master Agent. See "Starting the SNMP EMANATE Master Agent" on page 14.

5 Configure e-Biz Impact clusters to publish telemetry to the SNMP service. See "Configuring clusters to publish SNMP telemetry" on page 19.

6 Start the cluster. See "Starting clusters" on page 20.

7 Start the Global Console. See "Starting the Global Console" on page 20.

8 Create an SNMP agent for each cluster you want to monitor. See "Creating SNMP agents" on page 20.

Setting the *ims* wrapper script path

You must set the PATH environment variable for the *ims* scripts, which are used to execute some initial monitoring configuration commands. The following SNMP *ims* scripts are included with e-Biz Impact.

Script	Description
ims.sr	A general wrapper script for SNMP commands.
ims.setsrdir	Creates a directory for an SNMP port. You only need to run this command once. Example: `ims.setsrdir` *<port>* where *<port>* is the SNMP port to use.
ims.setsrports.sh	Sets port environment variables for the UNIX Bourne shell.

Script	Description
ims.setsrports.csh	Sets port environment variables for the UNIX C shell.

The ims executable is located in *x:\Sybase\ImpactServer-5_4\bin* on Windows and in *~/Sybase/ImpactServer-5_4/bin* on UNIX systems, where "*x*" and "~" is the directory, system, or drive where the e-Biz Impact 5.4.5 server is installed.

Windows

❖ **Setting the PATH variable on Windows**

1 Select Start | Settings, then double-click Control Panel.

2 When the Control Panel window opens, double-click the System icon.

3 When the System Properties window opens, select the Advanced tab, then click Environment Variables.

4 In the System Variables section of the window, select the PATH variable and click Edit.

5 In the Edit System Variable dialog box, append the *ims* wrapper script location to the PATH statement. For example, if the e-Biz Impact server is installed in the root of drive D: on Windows, enter this at the end of the PATH statement in the Variable Value field:

```
;D:\Sybase\ImpactServer-5_4\bin
```

Note Separate the existing path statement and the new entry with a semicolon (;) with no space after the semi-colon.

6 Click OK to save your entry and close the Edit System Variable dialog box.

7 Click OK to close the Environment Variable window, then click OK to close the System Properties window.

8 Close the Control Panel window.

9 Restart your machine to implement the new PATH statement.

UNIX

❖ **Setting the PATH variable on UNIX**

This procedure is for the C-shell user environment. If your UNIX system uses a different shell (for example, the Bourne shell or Korn shell), see your UNIX system documentation for instructions on setting the PATH variable.

Note To see which shell is the default on your system, type `echo $SHELL` in a terminal window.

You can set the PATH variable each time you open a terminal session and start the cluster, or you can set the PATH variable permanently for all terminal sessions by adding the ims wrapper script path to the *$HOME/.cshrc* file.

Note Generally, shell variables apply only to the current instance of the shell and are used to set short-term working conditions; environment variables have a farther reaching significance, and those set at login are valid for the duration of the session. By convention, environment variables are uppercase and shell variables are lower case names.

The PATH environment variable and path shell variable specify directories to search for commands and programs. Both variables always represent the same directory list, and altering either automatically causes the other to be changed.

---

1 To permanently add the *ims* executable to your path:

a Open the *$HOME/.cshrc* file in a text editor.

b Add the following line to the *.cshrc* after the list of other commands:

```
set path = ($path /<install_directory>/Sybase/ImpactServer-5_4/bin)
```

where <*install_directory*> is the location where the e-Biz Impact version 5.4.5 server is installed.

c Save the *.cshrc* file and close the text editor.

d Log out of the system and log back in to establish the new path.

2 To add the *ims* path to the end of your existing path for only the current session, open a terminal window and enter the following on one line:

```
set path = ($path <install_directory>/Sybase/ImpactServer-5_4/bin)
```

where <*install_directory*> is the location where the e-Biz Impact version 5.4.5 server is installed.

Note If you did not use step 1 to permanently add the *ims* wrapper script location to your path, each time you open a terminal window to execute the cluster, you must first enter the command shown in step 2 to set the PATH variable for the ims wrapper script for that session.

Configuring the SNMP port directory

Before you can run a SNMP daemon, you must configure the directory for the port the daemon uses. This must be done for the standard port of 161, as well as for any custom ports. Setting up the configuration directory is a one- time event, done before you run the daemon for the first time.

To set up the standard port, enter this command at a Windows command line or in a UNIX terminal window:

```
ims.setsrdir 161
```

Starting the SNMP EMANATE Master Agent

SNMP-based monitoring requires an agent. The SNMP EMANATE Master Agent is provided and installed with the e-Biz Impact product. Prior to starting the Global Console, the SNMP EMANATE Master Agent (snmpdm) must be running. If you run the e-Biz Impact server on Windows, the SNMP EMANATE Master Agent is configured as an automatic service. If you install the e-Biz Impact server on UNIX, the agent runs as a daemon.

Running multiple SNMP daemons

e-Biz Impact allows you to run multiple SNMP daemons on one system. To run multiple SNMP daemons, modify the Global Console agent configuration interface and its runtime.

Before you start the SNMP EMANATE Master Agent, set the environment variables *SR_SNMP_TEST_PORT* and *SR_TRAP_TEST_PORT* to match your cluster configuration. If the e-Biz Impact client is installed on Windows, the Global Console must use the same port that is used by the e-Biz Impact server.

Note Due to the limitations of the Microsoft API used for SNMP when using a nonstandard SNMP port, you can monitor the system using Global Console only from Windows 2000 or Windows XP.

If the service or daemon does not start automatically, start the service manually using the *ims.sr* wrapper script, or create a script to start the service or daemon at system startup.

Windows

- To install the SNMP EMANATE Master Agent as a Windows service:

  a Open a command prompt window.

  b Go to the root drive; for example, *C:* or *D:*.

  c At the command prompt, enter:

```
x:\Sybase\ImpactServer-5_4\bin\ims.sr snmpdm -install -tcpany
```

  where "*x*" is the drive where e-Biz Impact server is installed.

  The `-tcpany` flag allows a connection from any host and overwrites the default configuration, which allows TCP/IP connections only from the local host.

- To start the Windows service, enter this command in a command line window:

```
ims.sr snmpdm -start
```

  or

  - Select Start | Settings | Control Panel | Administrative Tools | Services. Right-click the SNMP EMANATE Master Agent service in the right pane and select Start.

- To start the daemon in debug mode, use the `-d` option:

```
ims.sr snmpdm -d -tcpany
```

Warning! Do not run the service in debug mode from a batch (*.bat*) or command (*.cmd*) file. The window may not display correctly after the process exits, making it difficult to troubleshoot errors.

- To uninstall the service, enter:

```
ims.sr snmpdm -remove
```

UNIX

On UNIX, you must run snmpdm as root to bind to the default SNMP port.

- To start the daemon, enter the following command from the root directory in a terminal window:

```
.../Sybase/ImpactServer-5_4/bin/ims.sr snmpdm -install -tcpany
```

where "..." is the directory where e-Biz Impact server is installed.

The `-tcpany` flag allows a connection from any host and overwrites the default configuration, which allows TCP/IP connections only from the local host.

- To start the daemon in debug mode, use the `-d` option:

```
ims.sr snmpdm -d -tcpany
```

Stopping SNMP

Windows

To stop the SNMP EMANATE Master Agent service:

- Enter this command in a command line window:

```
ims.sr snmpdm -stop
```

or

- Select Start | Settings | Control Panel | Administrative Tools | Services. When the Services window displays, right-click the SNMP EMANATE Master Agent service and select Stop.

UNIX

Find the process ID using the ps command, then use the kill command with the SIGKILL (9) signal. For example, enter:

```
kill -KILL <pid>
```

or

```
kill -9 <pid>
```

Where *pid* is the process ID that you identified from the ps command.

Configuring SNMP for non-standard port use

To specify non-standard ports for the daemon, use any available port to run the SNMP daemon.

Warning! Custom SNMP ports for the Global Console work only on Windows 2000 or XP. Due to operating system limitations, Windows NT can use only the standard port.

You might want to use a non-standard port because:

- Another SNMP daemon is running and you want to keep using it. This could be to support high availability.
- To isolate your production environment from your test environment.
- You do not have root access. The standard SNMP port is privileged, requiring you to be root.

For general configuration, do the following:

Set	To
SR_MGR_CONF_DIR	*<NNSY_ROOT>/snmp/srconf/mgr* on UNIX and *<NNSY_ROOT>\snmp\srconfmgr* on Windows, where *<NNYSY_ROOT>* is the e-Biz Impact server installation directory; for example, *working/Sybase/ImpactServer-5_4/*.
SR_AGT_CONF_DIR	*<NNSY_ROOT>/snmp/srconf/agt* or *<NNSY_ROOT>\snmp\srconfagt*
SR_SNMP_TEST_PORT	The port number to use.
SR_TRAP_TEST_PORT	The port number used for traps.

Note If *SR_TRAP_TEST_PORT* is not set, it defaults to a value of one greater than the SNMP port number. For example, if *SR_SNMP_TEST_PORT* is set to 5000 and you do not set *SR_TRAP_TEST_PORT*, it defaults to 5001.

Setting environment variables

Windows NT

Because you set global environment variables to configure the SNMP daemon on Windows NT, you can run only one daemon as an NT service.

Windows 2000 and XP

You can set the *SR_MGR_CONF_DIR*, *SR_AGT_CONF_DIR*, *SR_SNMP_TEST_PORT*, and *SR_TRAP_TEST_PORT* environment variables on a per user basis and then configure the daemon to run as that user using the the Windows Services dialog box. When the daemon runs, it picks up its configuration from the user's environment. Only one SNMP EMANATE Master Agent can be run on Windows as a service. You can run other agents from a command prompt, setting the environment for the appropriate ports.

To set environment variables, log in to the system as the user. Environment variables are set under System Properties, which are in the Advanced tab for Windows 2000.

After installing the SNMP daemon as a service, you must modify the properties of the services, using the Services applet, to specify the user to run as. On Windows 2000, this in on the Log On tab under Properties. From the Log On tab, specify the user and password.

UNIX

Set the environment variables before you run the SNMP daemon (snmpdm).

- If you use the Bourne shell, use *ims.setsrports.sh* to set the two port environment variables, *SR_SNMP_TEST_PORT* and *SR_TRAP_TEST_PORT*. The *ims.setsrports.sh* script takes the two ports as command line arguments. For example, to use 5161 for the SNMP port and 5162 for the trap port, enter:

  ```
  . ims.setsrports.sh 5161 5162
  ```

- If you use the C shell, save the *ims.setsrports.csh* script to another name; for example, a name that matches the port values or matches the purpose for which the values are used. You can then modify the values that are set in the script to the values that you want to use.

An *.AgentSockets* directory is created in */tmp* that contains the socket devices that go with the UNIX domain sockets that are used by the master agent to communicate with subagents. If the daemon was started as root, it only has write permissions for root. Then you must open the permissions with the chmod command, for example: `chmod 777 /tmp/.AgentSockets`. If this is not acceptable, run the daemon as root to give the user access to the directory.

To run any of the SNMP commands distributed with e-Biz Impact, use the *ims.sr* shell script in the *../ImpactServer-5_4/bin* directory to ensure that the environment is set up correctly.

To run commands, pass the command as an argument to the script.

Changing SNMP settings

SNMP daemon settings for *MAX_THREADS* and *MAX_SUBAGENTS* can affect Global Console operation. To change these values, use a text editor to open *x:\Sybase\ImpactServer=5_4\srconf\agt.<port>\snmpd.cnf* on Windows and *~/Sybase/ImpactServer-5_4/snmp/srconf/agt.<port>/snmpd.cnf* on UNIX, where "*x*" and "~" represent the file system, drive, or directory where the e-Biz Impact server is installed.

MAX_THREADS

The default *MAX_THREADS* setting is 10. This is the number of worker threads used by the master agent running on the port. Worker threads satisfy requests for information from the Global Console, other clusters involved in cluster-to-cluster communications, and CNC requests (ims54cnc). The more threads that are being used, the more the maximum threads number must increase. If you add another Global Console and experience unexpected behavior, increase the number of *MAX_THREADS*.

MAX_SUBAGENTS

The default *MAX_SUBAGENTS* setting is 25. This is the number of subagents that the SNMP daemon supports. Each cluster process, which consists of the manager and each controller, is a subagent of the SNMP daemon. You should allow for subagents for cluster-to-cluster communication and CNC requests. For example, if this value is set too low, you could see only a partial list of what you would expect to display in the Global Console.

Configuring clusters to publish SNMP telemetry

To enable clusters to publish telemetry to the SNMP service:

1 Select Start | Programs | Sybase |e-Biz Impact 5.4 | Configurator.

2 When the Configurator window opens, right-click the e-Biz Impact Configurator node in the tree view and select Load Cluster.

3 When the Open dialog box appears, navigate to the cluster you want to configure, select it, then click Open.

4 Right-click the cluster in the tree view and select Properties.

5 When the cluster properties window opens, select the General tab, select the Publish Telemetry to SNMP Service option in the Monitoring section, then click OK.

6 Right-click the cluster in the tree view and select Save to save the cluster with the same name.

7 Repeat steps 2 through 6 for each cluster that you want to monitor in the Global Console.

8 Select Console | Exit to close the Configurator.

Starting clusters

To start a cluster, enter the following command at a Windows command prompt or in a UNIX terminal window:

```
ims cluster -cluster.name myCluster
```

where *myCluster* is the name of the cluster you want to run.

See the *e-Biz Configuration Guide*, Chapter 5, "Deploying Files and Executing e-Biz Impact Clusters" for more information about the command options for starting clusters.

Starting the Global Console

Note The Global Console is available only on Windows systems.

To start the Global Console, select Start | Programs | Sybase | e-Biz Impact 5.4 | Global Console.

The main Global Console window displays, which is a snap-in to the Windows Microsoft Management Console (MMC). See your Windows documentation for more information about using the MMC.

Note Global Console windows are designed to display at an 800 x 600 resolution. If all the information does not display in the right pane, maximize the window.

Creating SNMP agents

Create a Global Console agent for each e-Biz Impact server you want to monitor. Each agent uses SNMP to monitor all clusters associated with a specified e-Biz Impact server.

1 In the Global Console main window, right-click e-Biz Impact 5.4 Global Console in the tree view and select New | Agent. A new agent icon displays in the tree view and in the right pane.

2 When the Agent Properties window opens, complete these options on the General tab:

- Agent Name – enter a unique name for this agent, or accept the default. By default new agents are named Agent1, Agent2, Agent3, and so on. Accept the default or enter a different name.
- Host Name – specify the name of the host on which the e-Biz Impact server is running.
- SNMP Community String – the string of the SNMP service running on the e-Biz Impact server. The default value is "public." Change this value only if the value for the server was also changed.
- Refresh Rates – the refresh rate, in seconds, at which the Global Console polls data from the SNMP service.
- SNMP Port – the port on which the e-Biz Impact server is publishing SNMP. This port allows you to run multiple SNMP services on the same server (Window 2000 or later) without conflict.
    - Default – select this option to use the default SNMP port number, which is 161.
    - Custom – select this option to use a port other than the default. After you select this option, enter the port number in the provided field.
- Automatically Refresh After Executing A Command – select this option to refresh the console after a user executes a command.
- Reconnect at Startup – select this option to automatically reconnect this agent when the Global Console starts.

3 Click OK to save your entries and close the window.

When you create an agent, by default the agent monitors all cluster objects. However, after you save a new agent, you can modify the agent's properties and specify which objects to monitor.

Changing agent properties

The agent configuration that you set up when you first create an agent, monitors all clusters for a specified server, and all objects for those clusters. However, once an agent is created, additional properties are available that allow you to designate specific cluster components to monitor if you prefer not to monitor everything.

To view or change agent properties, right-click the agent icon in the tree view and select Properties. The agent properties window opens. In addition to the General tab, you now see Display, SNMP Audit, and Advanced tabs.

Note When you modify existing agent properties, the agent must not be connected to the e-Biz Impact server. Right-click on the agent in the tree view and select All Tasks | Disconnect.

General tab

The options available on the General tab are the same options that you complete when you create the agent. Modify the options as necessary, but be sure that the agent is not connected to the e-Biz Impact server when making changes. See “Creating SNMP agents” on page 20 for field descriptions.

Display tab

1 Select the Display tab.

You can monitor only problematic servers, or select specific objects to monitor. The system administrator who predefines the console for operators can create and connect the agent, and use the Display tab to hide objects. When the operator loads the console file with an agent automatically connected, the Global Console displays only the selected objects.

- All problematic objects only – select this option to display and monitor only problematic servers. This option may increase Global Console performance.

- Objects selected below – select this option for each object you want to display and monitor. Initially, and when the agent is not connected, no objects are available. The first time that the agent discovers the structure of the server it monitors, the list is filled with all discovered objects, from which you can make selections.

2 Click Apply, then click OK.

SNMP Audit tab

This tab displays what was published and successfully parsed by the Global Console. The Global Console uses the data to build the tree structure in the main window tree pane. Each object, and its associated properties acquired from SNMP, displays in the Global Console right pane.

Advanced tab

1 Select the Advanced tab, which provides alert options and advanced properties that you can set to accommodate slower networks.

- SNMP Session Parameters –
  - Max Number of Retries – enter the maximum number of times to poll the SNMP connection for data.
  - Request Timeout (ms) – enter the amount of time, in milliseconds, to wait when polling the SNMP connection.
- Play audio file when agent is healthy – when you select this option, enter the name of the audio *.wav* file to use, or click the ellipses button to navigate to and select the file from a dialog box.

Note You can also enable and disable this option from the Global Console main window: right-click an agent in the tree view or right pane and select All Tasks | Healthy Sound | [Enable/Disable].

- Play audio file when agent is problematic – when you select this option, enter the name of the audio *.wav* file to use, or click the ellipses button to navigate to and select the file from a dialog box.

  Note You can also enable and disable this option from the Global Console main window: right-click an agent in the tree view or right pane and select All Tasks | Problematic Sound | [Enable/Disable].

- Display SNMP errors in a message box instead of the status bar – select this option to display SNMP errors in a separate message dialog box.

2 Click OK to save your entries, exit, and close the window.

You are now ready to begin monitoring e-Biz Impact cluster activities. See Chapter 3, “Using the Global Console.”

CHAPTER 3

Using the Global Console

This chapter covers the operations of a Global Console as previously defined by a system administrator. If Global Console operations have not been defined, see Chapter 2, “Configuring the Global Console.”

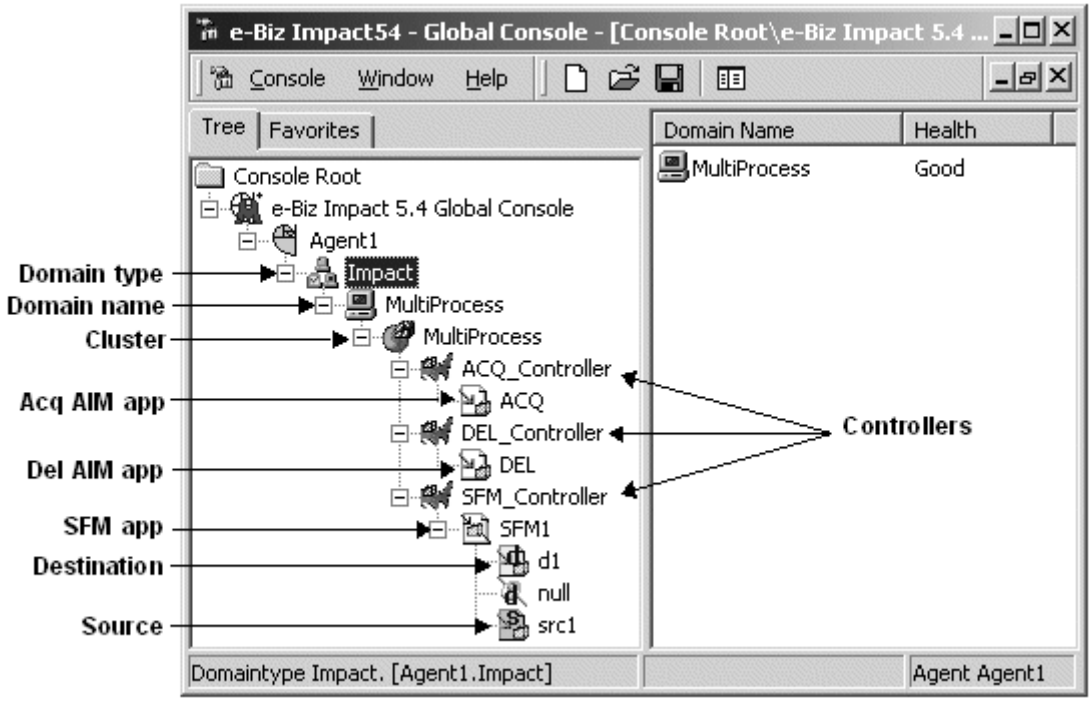
Introduction

When you create an agent in the Global Console, and establish an agent connection to an e-Biz Impact server, the agent builds a list of nodes in the Global Console tree view that represents the bus structure published by the server. The e-Biz Impact bus structure can include one or several:

- Domain types (default—Impact)
- Domain names (default—Impact)
- Clusters
- Controllers
- Applications
- Sources
- Destinations

Figure 3-1 illustrates a sample bus structure shown in the tree view of Global Console.

Figure 3-1: Global Console



The bus structure reflects the server configuration that has been defined in the Configurator (see the *e-Biz Impact Configurator Guide*). The domain type and domain name nodes are configured at runtime and set when the server starts. These nodes allow an agent to monitor multiple instances of the same cluster. For example, you can run development, test, and production environments all on the same host.

A small green check mark in the lower left corner of a tree view icon indicates the currently selected node. In Figure 3-1, the currently selected object is the Impact domain type node.

General server state

To display the overall health of each e-Biz Impact server, select the e-Biz Impact 5.4 Global Console node in the tree view to see the server's general state indicated in the right pane by color. When the right pane is green, everything is functioning normally. When the right pane is red with a flashing black X, maintenance is required.

Object health

The Global Console displays visual and text cues to indicate the health of each cluster object. When you select any object in the tree view except an application beneath a controller object, the health of the selected object is indicated by its icon in the tree view and the text that displays in the Health column of the right pane. Health is defined as:

- Good – the object operates normally.
- Problematic – a problem associated with any children objects of a node. All icons above a problem node in the tree view are marked with a red arrow and "Problematic" displays in the right pane's Health column.
- Bad – there is a problem associated with a node. A red circle on the object's icon and "Bad" in the right pane's Health column identifies an unhealthy object.

These visual and text cues allow an operator to visually locate unhealthy objects in the bus by looking at the tree view nodes.

Note The Global Console can be configured to display only problematic clusters. See "Changing agent properties," the subsection "Display tab" on page 22 for instructions.

Object lists

When you select a node in the tree view, the right pane displays a list of the child objects of the selected node. For example, in Figure 3-1, the Impact domain type selected in the tree view displays the MultiProcess domain name in the right pane. If you selected the MultiProcess cluster in the tree view, the ACQ_Controller, DEL_Controller, and SFM_Controller would be listed in the right pane.

To sort any column in the right pane, click the column header. For example, if you click the Controller column, all controllers are sorted in alphabetical order.

Application views

Controller applications and SFM or router destinations and sources have multiple views. Each view highlights particular information in the right pane. The view is refreshed at the Refresh Rate value (in seconds) specified on the General tab of an agent's properties. See "Changing agent properties" on page 22.

To access object views, right-click the small green check mark in the lower left corner of the application's icon in the tree view and select View. You can also select the application in the tree view, then select View from the Global Console menu bar.

Note If the View option is not on the menu when you right-click an object in the tree view, select View from the Global Console menu bar to confirm whether the object has views available.

If Global Console views are not available, the View menu or the right-click View menu option displays the standard Windows view selections (Choose Columns, Large Icons, Small Icons, List, Detail, Customize).

See "Displaying ODL AIM application views" on page 30

Global Console tasks

When you monitor e-Biz Impact clusters, you have the opportunity to interact with some of the transactions views. Clusters, SFMs, sources, and destinations all provide a variety of commands that the operator can invoke.

Right-click a cluster, SFM, source, or destination and select All Tasks to see the commands that can be executed for the selected object.

See "Performing Global Console tasks" on page 49 for more information.

Connecting and logging in to an e-Biz Impact server

To start the monitoring process, connect the agent and log in to the e-Biz Impact server where the clusters that you want to monitor reside.

Once a connection is established, the agent monitors the e-Biz Impact server by representing the data received from the server MIB (Management Information Base) through SNMP.

Note The MIB is similar to a schema for published data. The MIB describes how the data is published; for example, organization and data types. The e-Biz Impact MIB file is *Impact8.my*, which is located in *x:\Sybase\ImpactServer-5_4\snmp* on Windows and in *~/Sybase/ImpactServer-5_4/snmp/* on UNIX, where "*x*" and "~" represent the drive, directory, or file system where the e-Biz Impact server is installed.

❖ **Connecting an agent to an e-Biz Impact server**

1 In the Global Console on Windows, select the agent in the tree view.

2 Right-click the selected agent and select All Tasks | Connect.

3 To automatically connect an agent when you start the Global Console, select the Reconnect at Startup option on the agent properties General tab.

When an agent connection is established, the e-Biz Impact server bus structure displays in the main window.

❖ **Logging in to a e-Biz Impact server**

To execute commands on the applications contained in the cluster, you must log in to the e-Biz Impact server.

1 Right-click the cluster icon in the tree view and select Login.

2 Enter the user ID and password associated with a user that is defined in the Authorization database.

If the agent is unable to connect, an error appears, indicating the source of the problem. Check to see if the agent properties, such as community string or port number, are valid, and verify that SNMP service is started on the server.

Note If you are using e-Biz Impact security, operators can execute only the commands for which their security role is authorized. See the *e-Biz Impact Authorization Guide* for more information. See the *e-Biz Impact Command Line Tools Guide* for more information about CNC requests.

Monitoring e-Biz Impact

Note SNMP daemon settings for *MAX_THREADS* and *MAX_SUBAGENTS* can affect Global Console operation. See the "Changing SNMP settings" on page 19 for instructions on change these settings.

When an agent connection is established, the e-Biz Impact server bus structure displays in the main window (Figure 3-1).

This section describes all available Global Console view. To review CNC requests that authorized users can execute from Global Console, see "Performing Global Console tasks" on page 49.

Displaying ODL AIM application views

Global Console provides views for ODL acquisition and delivery AIM applications. Each ODL application corresponds to a thread in the controller process. These applications execute ODL code at runtime and acquire or deliver transactions.

Each ODL AIM application registers its telemetry properties via its parent controller, which acts as a subagent. The subagent gathers various application telemetry properties when instructed by the SNMP EMANATE Master Agent.

❖ **Displaying ODL AIM application views**

1 In the Global Console tree view, select an application or delivery AIM beneath a controller.

2 Select View on the Global Console menu bar.

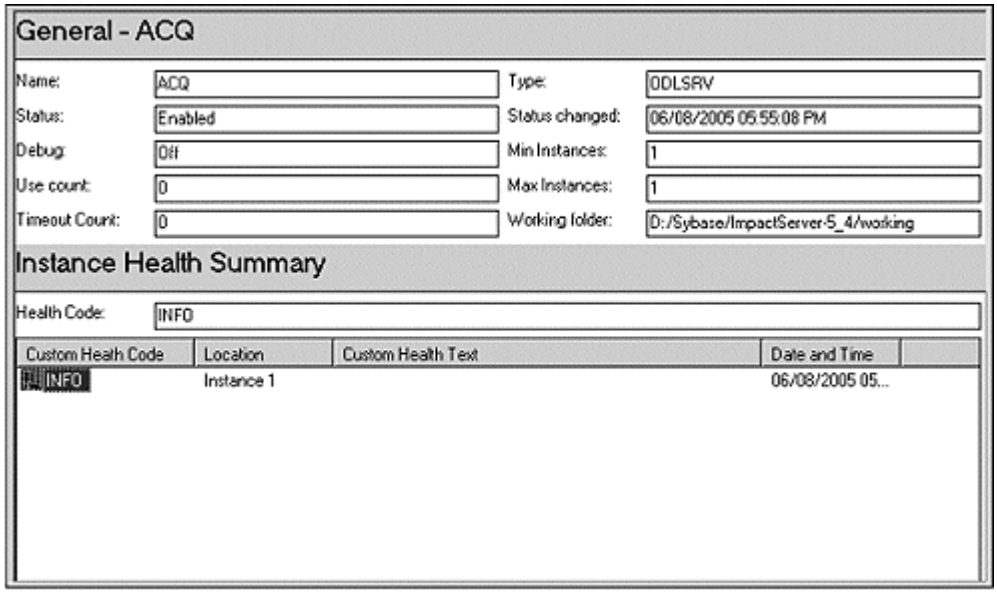
Note You can also right-click the small green check mark in the lower left corner of the application's icon in the tree view and select View.

3 Select an available view—General, Instances, and Functions.

General view

The General View (Figure 3-2) allows an operator to view general information about an ODL AIM application.

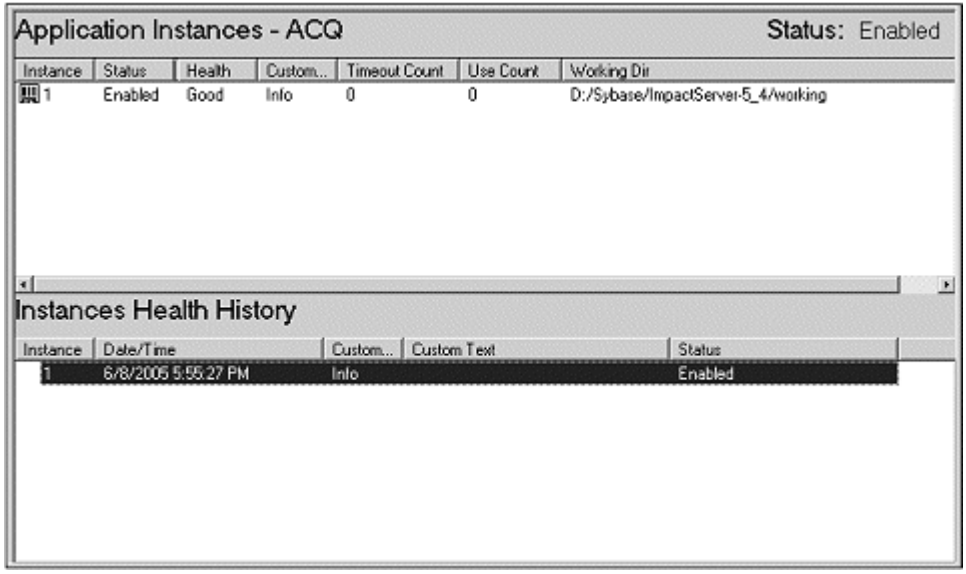
Figure 3-2: ODL AIM application General view



Instances view

The Instances view (Figure 3-3) displays the number of instances of the application currently enabled. An application can be configured to run more than one instance. Each instance is an independent copy of the application.

Figure 3-3: ODL AIM application Instances view



Beneath Instances Health History, application instances can display a Custom Status and Custom Text, which allow the AIM developer to raise errors or warnings inside the ODL code.

Custom Status values are:

- Info – corresponds to “Good”
- Warning – corresponds to “Problematic”
- Error – corresponds to “Bad”

Changing application health status within ODL

ODL publishing allows developers to change an object’s health according to the conditions shown. For example, when the endpoint application fails.

To publish an AIM’s status and text from ODL, call this function:

```
clSetAimStatus(int code,
string *text);
```

where *code* displays in the Instance Health Summary section of the right pane as Health Code:

- INFO (good)
- WARNING (problematic)

- ERROR (bad).

and *text* is user-defined content that appears in the Custom Health Text column of the Instance Health Summary section in the bottom pane of the General view.

Note For more information about clSetAimStatus(), refer to the *e-Biz Impact ODL Guide*.

By default, an endpoint failure does not change the status of the e-Biz Impact destination to which is sending. From an e-Biz Impact point of view, the destination is still alive and is capable of accepting transactions from the SFM, so to the console, the destination appears healthy. ODL publishing allows the user to set the destination's health to “bad” if this condition occurs.

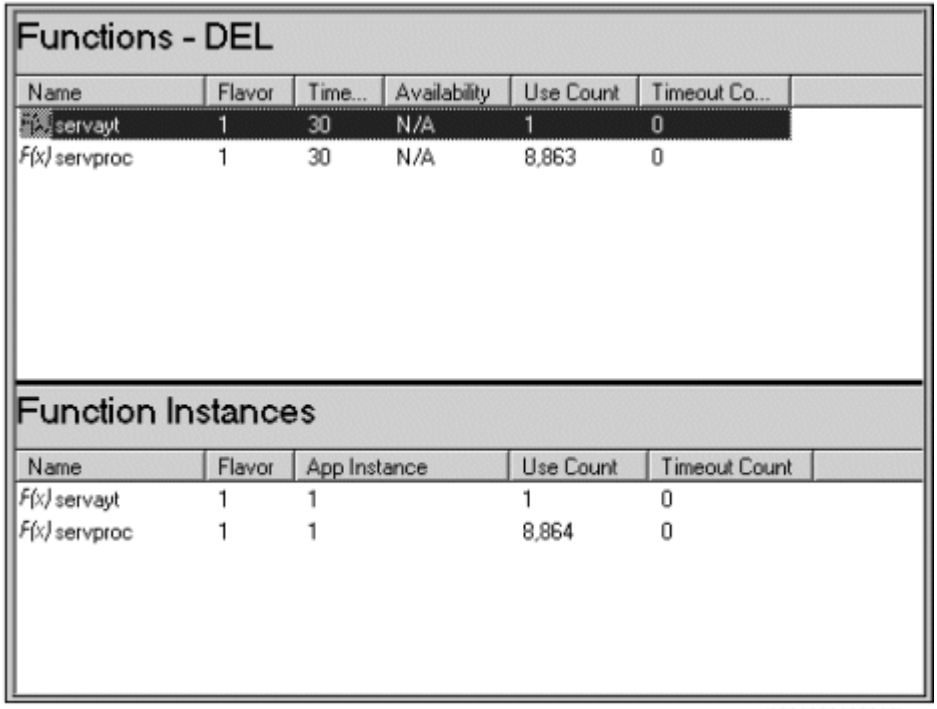
ODL supports publishing an AIM’s status with text to the Global Console. This information shows up in the Global Console as Custom Health Code and Custom Health Text.

Custom Health Text is a user-defined message written in ODL and raised by an application.

Functions view

The Functions view (Figure 3-4) displays the available ODL AIM application functions, as well as the number of instances that currently use the functions.

Figure 3-4: ODL AIM application Functions view



Each e-Biz Impact application executes ODL code from files that are deployed in an application's working directory. These files prototype and implement several functions used at runtime to perform specific transaction operations. The application publishes its functions and tracks the number of times functions are used.

Displaying SFM application views

An SFM application is similar to a regular ODL application, but has enhanced capabilities. These capabilities require a special views for tracking and managing transactions.

❖ **Displaying SFM application views**

1 In the Global Console tree view, select an SFM application beneath an SFM controller.

2 Select View on the Global Console menu bar.

Note You can also right-click the small green check mark in the lower left corner of the SFM application's icon in the tree view and select View.

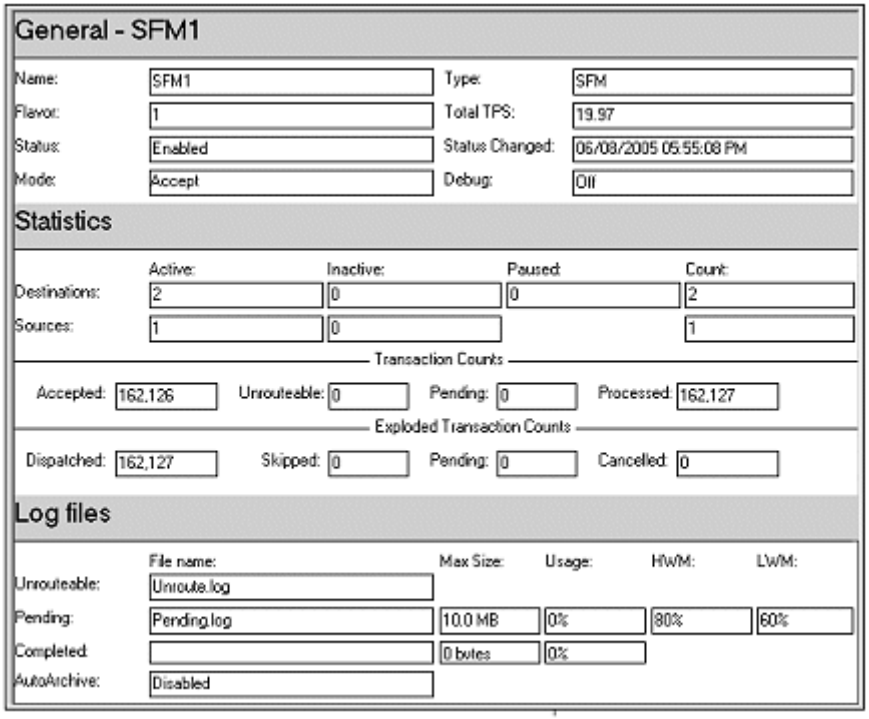
3 Select an available view—General, Functions, Sources and Destinations, Transactions Graph, Dashboard, Unrouteable Transactions, Pending Transactions, Unprocessable Transactions, and Cancelled Transactions.

SFM applications do not have an Instances view. Transactions are polled and processed by "first in, first out" mode.

General view

The SFM General view (Figure 3-5) allows an operator to view general information about an SFM application, including transaction count statistics and high/low watermarks. The General view also provides in-depth statistics, and indicates situations that affect the SFM's health.

Figure 3-5: SFM application General view



Note You can also view general information about an SFM's sources and destinations. Select a source or destination beneath an SFM in the tree view, then select General from the MMC View menu. See "SFM source General view" on page 45 and "Destination General view" on page 46.

Functions view

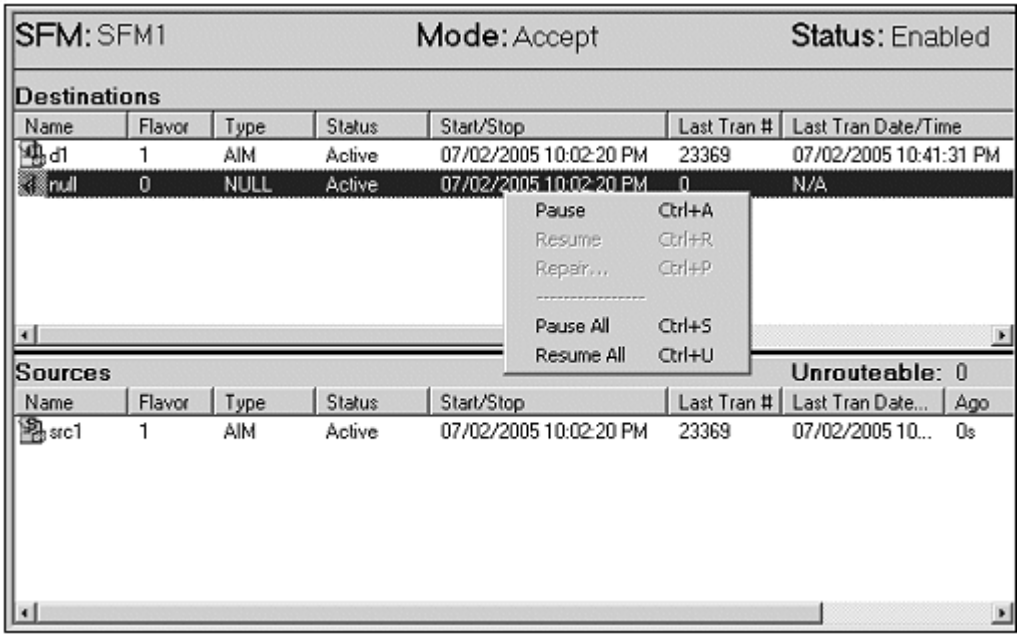
The SFM application Functions view displays same information as the ODL AIM applications Function view. You see the available SFM application functions, as well as the number of instances that currently use the functions.

Select View | Functions. The right pane displays information similar to the ODL AIM application Function view example shown in Figure 3-4.

Sources and Destinations view

The Sources and Destinations view (Figure 3-6) displays available information for the sources and destinations used by the associated SFM. When logged in to the e-Biz Impact server, an operator can execute CNC request operations on the destinations (right-click the pause, resume, pauseAll, resumeAll).

Figure 3-6: SFM application Source and Destination view



These views offer telemetry properties that are useful when an operator wants to track the internal activities of all sources and destinations that acquire or deliver transactions into or out of an SFM. Telemetry properties include:

- Number of transactions pending, skipped, or cancelled for each destination queue
- Length of time since the last transaction was acquired and delivered
- Overall and individual health

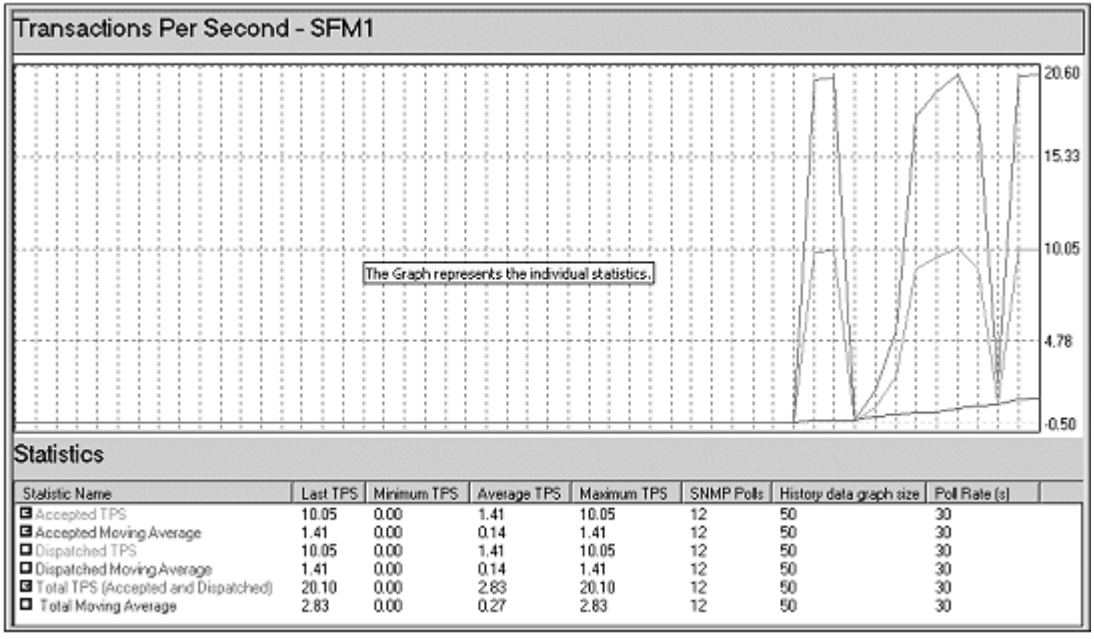
Transaction Graph view

The Transaction Graph view (Figure 3-7) represents the workload for the selected SFM application. This performance information can help you plan for the best implementation of your Impact servers and to optimize and tune the servers already in place.

The Transaction Graph view provides the following static graphs as well as graphs depicting moving averages for each:

- Accepted transactions
- Dispatched transactions
- Total TPS (SFM workload)

Figure 3-7: SFM application Transaction Graph view



Statistic Name	Last TPS	Minimum TPS	Average TPS	Maximum TPS	SNMP Polls	History data graph size	Poll Rate (s)
Accepted TPS	10.05	0.00	1.41	10.05	12	50	30
Accepted Moving Average	1.41	0.00	0.14	1.41	12	50	30
Dispatched TPS	10.05	0.00	1.41	10.05	12	50	30
Dispatched Moving Average	1.41	0.00	0.14	1.41	12	50	30
Total TPS (Accepted and Dispatched)	20.10	0.00	2.83	20.10	12	50	30
Total Moving Average	2.83	0.00	0.27	2.83	12	50	30

Reading the transaction graph

An SFM, a router, and a destination offer a transaction-per-second graph that represents the workload of the specified object. Use the check boxes to display or hide graph lines for various activities.

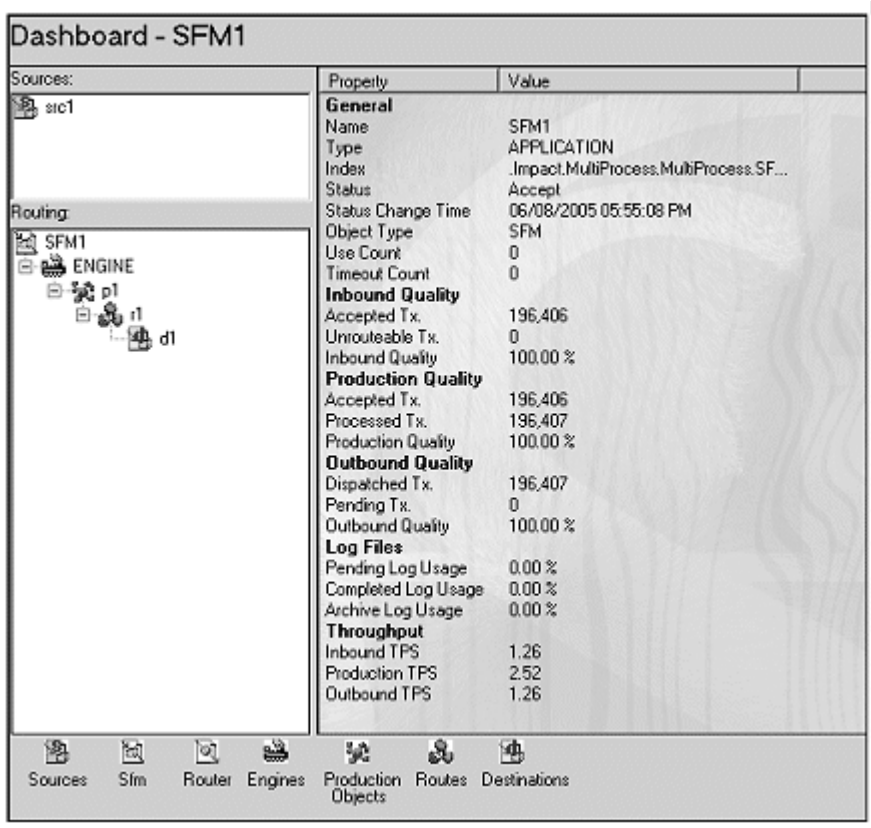
History length is fixed at 50 polls because the agent must have a constant poll interval, which allows the TPS to function. When the agent poll interval changes, the TPS is recalibrated.

Note You can also view a transaction graph an SFM's destinations. Select a destination beneath an SFM in the tree view, then select Transaction Graph from the MMC View menu. See "Destination Transaction Graph view" on page 47.

Dashboard view

The Dashboard view provides displays the internal architecture of an SFM by creating a hierarchical tree that represents the relationship between the SFM's engines, production objects, routes, and destinations.

Figure 3-8: SFM application Dashboard view



Select an object in the Sources or Routing view panes to display telemetry computations such as the percentage of unrouteable transactions, percent of cancelled transactions, percent of pending log file usage, and average input TPS.

Unrouteable Transactions view

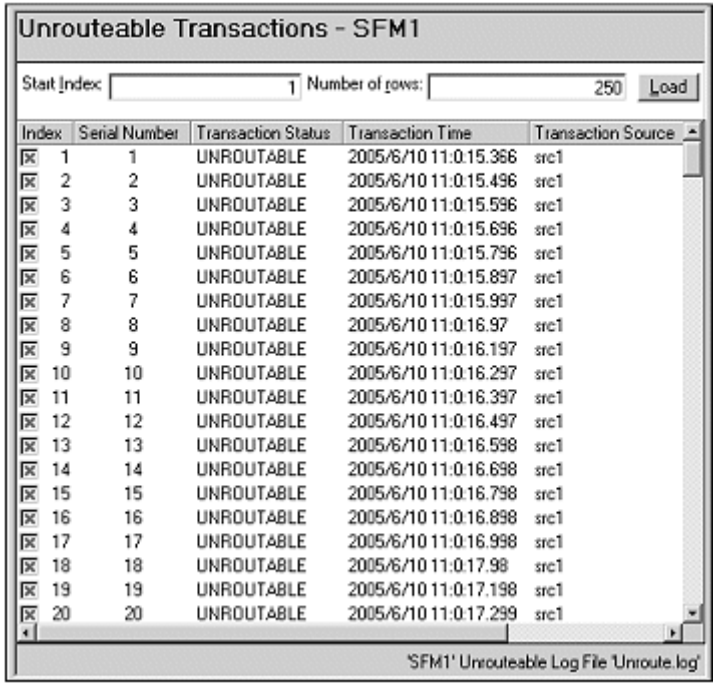
The Unrouteable Transactions view (Figure 3-9) displays transactions that do not qualify for any SFM production objects, and, therefore, are not routeable to any destinations. These transactions are stored in the unrouteable log file, which is named *Unroute.log* by default.

Note To change log file names and locations, right-click an SFM in the Configurator, select Properties, then select the Log Files tab.

The Unrouteable Transactions view allows you to edit unrouteable transactions and resubmit corrected transactions to the SFM.

You can also skip and repair unrouteable transactions. See “Repairing unprocessable, cancelled, and unrouteable d transactions” on page 58.

Figure 3-9: SFM application Unrouteable Transactions view



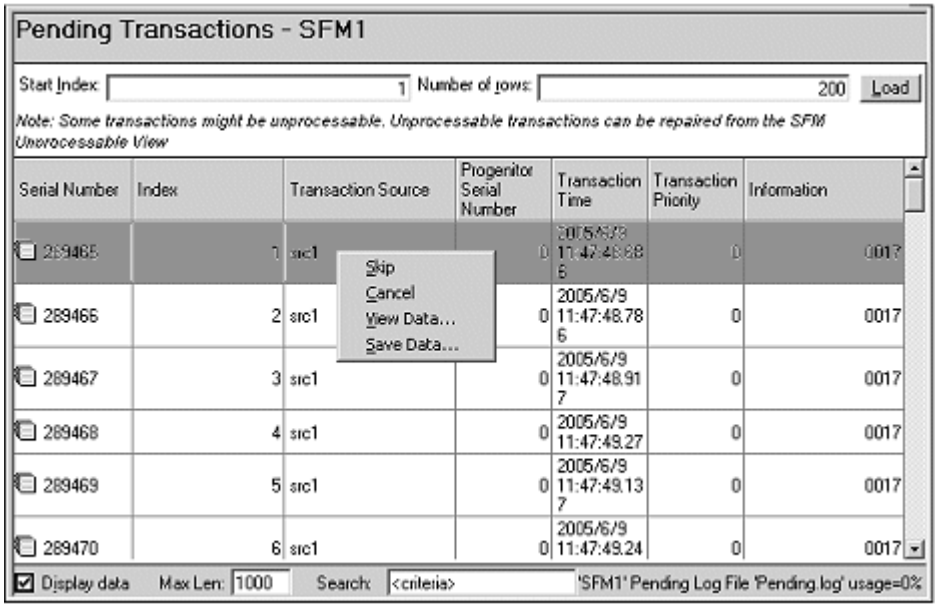
To delete a transaction, right-click the transaction and select delete. The transaction is not removed from the unrouteable log file, but is marked as DELETED. You cannot repair deleted transactions.

When a transaction is repaired, the SFM generates a new transaction that is linked to the original transaction serial number, which prevents loss of the initial transaction and provides a repair history. Once the transaction is repaired, the original transaction is given a REPAIRED status.

Pending Transactions view

The Pending Transactions view (Figure 3-10) lists transactions that are queued and in process by the SFM. Right-click a pending transaction to skip or cancel the transaction. The transaction index indicates position of a transaction in the processing queue.

Figure 3-10: SFM application Pending Transactions view



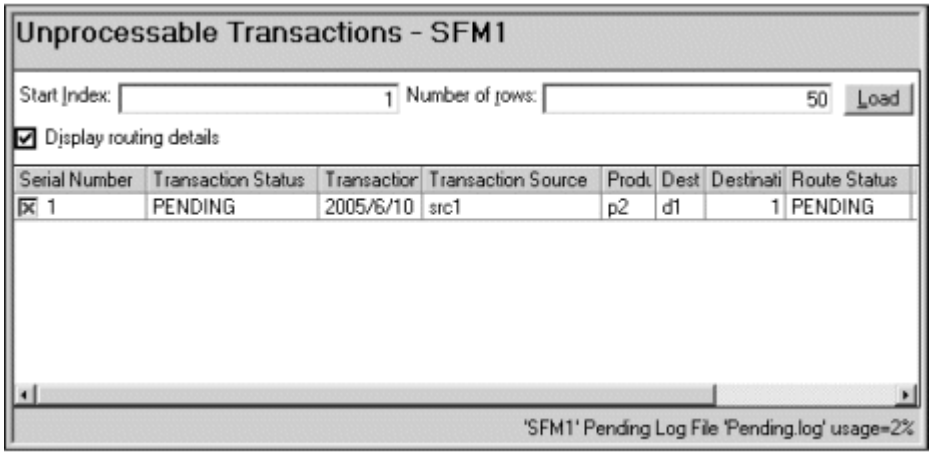
Unprocessable transactions are considered Pending and display in Pending Transactions view. For advanced operation on these transactions, see "Unprocessable Transactions view" on page 42.

Note You can also view pending transactions for an SFM's destinations. Select a destination beneath an SFM in the tree view, then select Pending Transactions from the MMC View menu. See "Destination Pending Transactions view" on page 48.

Unprocessable Transactions view

The Unprocessable Transactions view (Figure 3-11) displays transactions that fail to translate before being dispatched to their intended destination. Because a transaction can qualify for more than one production object, a transaction can fail in the translation phase for one or some of the production objects for which it qualifies. When only part of a transaction fails to translate, other instances of the transaction that qualify for other production objects are dispatched.

Figure 3-11: SFM application Unprocessable Transactions view



Use this view to correct unprocessable transactions by editing the transaction data and resubmitting the corrected transaction to a specific route.

When a transaction is being dispatched, one translation is performed on the transaction for each intended destination. For example, a transaction qualifies for a production object, which has three destinations (Dest1, Dest2, and Dest3). During dispatch to Dest1, translation is performed and the result is sent, just as for Dest2 and Dest3. When a transaction fails to translate en route to a destination, the internal dispatch queue for that destination is disabled. Until the failed transaction is fixed or removed, transactions lined up behind it also fail unless they have a higher priority than the first failed transaction.

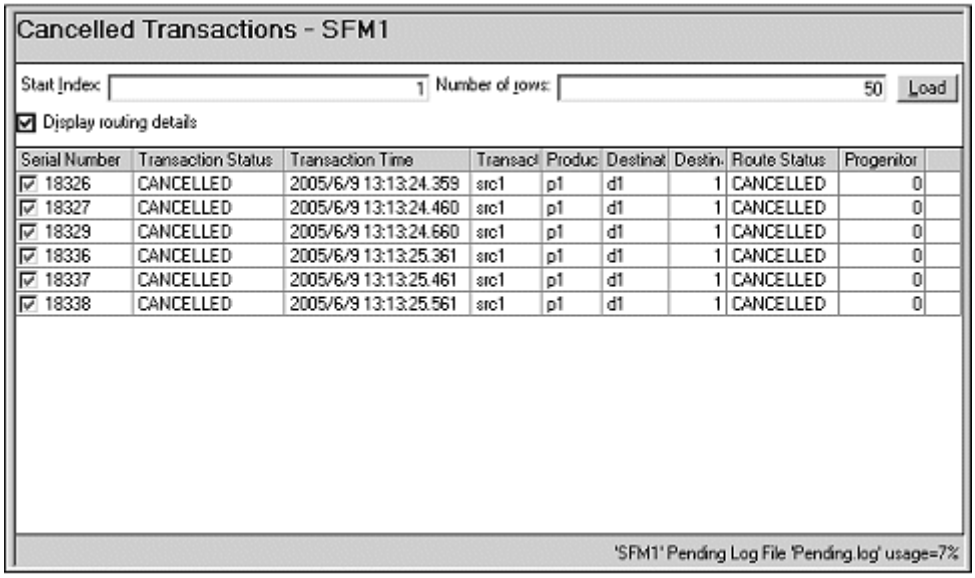
Unprocessable transactions can be cancelled or skipped to allow the SFM to continue processing data. Cancelled transactions are still considered in process; therefore, these transactions require that an operator either skip or uncancel them for repair. See “SFM Cancelled Transactions view” on page 57 and for more information. Select Routing Detail to view the status of the transaction on each individual route.

You can also skip and repair unprocessable transactions. See “Repairing unprocessable, cancelled, and unrouteable d transactions” on page 58.

Cancelled Transactions view

The Cancelled Transactions view (Figure 3-12) lists transactions that have been cancelled by the end point application or an operator. Uncancelled operations are given a PENDING status and reprocessed later by the SFM.

Figure 3-12: SFM application Cancelled Transactions view



Right-click a transaction's serial number and select Uncancel, Skip, or Repair for the selected transaction. See "Executing SFM tasks" on page 53.

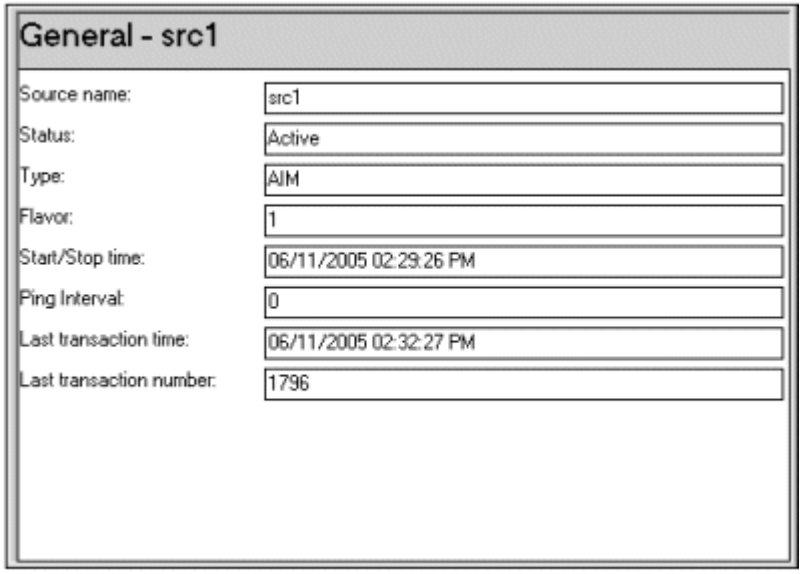
Displaying source views

Sources reference ODL application instances that make route_vrec, route_vprod, or routev_eng calls, make DFC calls, and distribute transactions to an SFM application.

General view

Select an SFM source in the tree view, then select General from the MMC View menu. You see general information about sources, similar to Figure 3-13. Each source publishes telemetry related to its context while transferring data to the SFM. ODL applications used as a source for an SFM can implement the ping DFC function.

Figure 3-13: SFM source General view



Displaying destination views

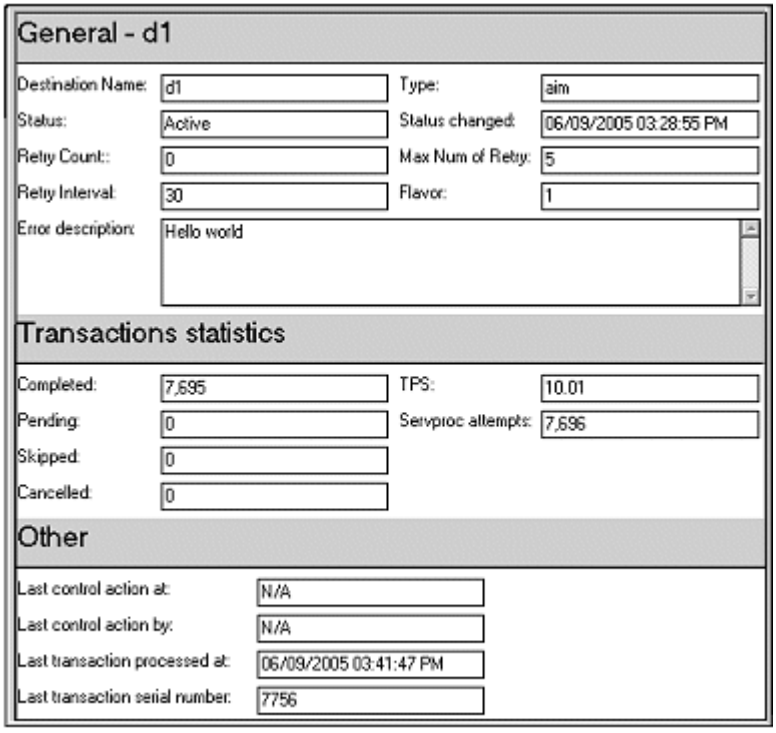
A destination references an ODL application that implements the basic DFCs servproc and servayt through which an SFM dispatches transactions. Destinations are responsible for handling a connection with the destination endpoint system, distributing transactions, and returning information about the dispatch to the SFM.

Upon a transaction's successful delivery, the SFM removes the transaction from the destination queue.

General view

To monitor SFM destinations, select an SFM destination in the tree view, then select General from the MMC View menu. A destination's General view (Figure 3-14) provides general information, transaction statistics, and error descriptions.

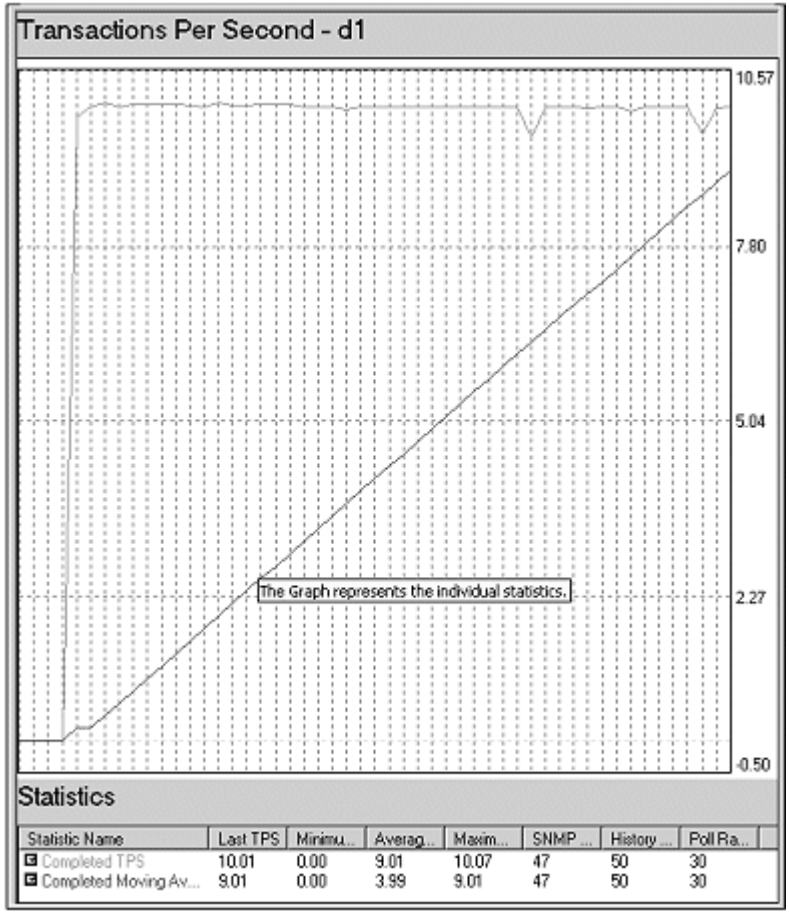
Figure 3-14: Destination General view



Transaction Graph view

To view a destination's transaction graph (Figure 3-15), select a destination below an SFM in the tree view, then select Transaction Graph from the MMC View menu. The transaction graph represents the number of dispatched transactions per second to the destination.

Figure 3-15: Destination Transaction Graph view



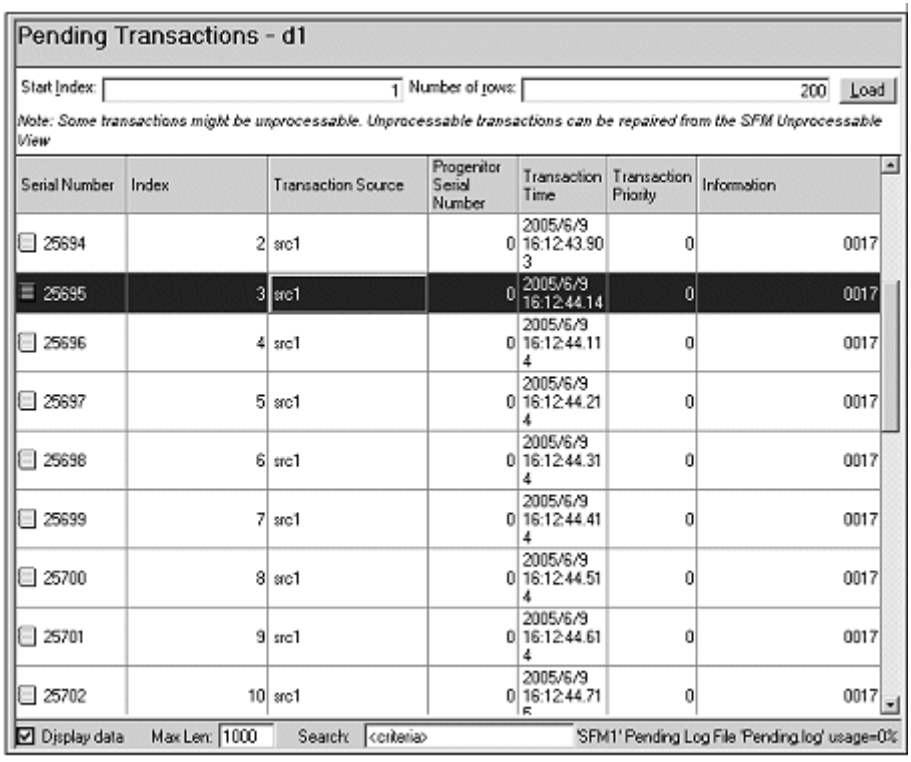
Graph history size is fixed at 50 polls. For accurate representation, the agent poll must be constant. If the poll number is modified, the graph recalibrates.

Pending Transactions view

This view lists pending transactions for a specific destination, select a destination below an SFM in the tree view, then select Pending Transactions from the MMC View menu. You see information display in the right pane similar to the example shown in Figure 3-16.

After translation, output transactions are queued to a destination, awaiting end point distribution. The Pending Transaction view for a destination allows you to interact with the queued output transaction prior to its delivery.

Figure 3-16: Destination Pending Transactions view



Right-click a transaction's serial number to skip, or cancel a transactions and view or save the selected transaction's data. The view does not show the content of the output transaction, but references the initial transaction. Transactions cannot be modified after translation. See "SFM Cancelled Transactions view" on page 57.

Performing Global Console tasks

Note If you are using e-Biz Impact security, the operator's authorization role controls which tasks that operator can perform. See the *e-Biz Impact Authorization Guide* for details. If you are not using security, any operator can execute any command.

The Global Console allows an operator to execute command and control (CNC) requests on clusters and SFM objects. When an operator or administrator executes a command in the Global Console, a set of CNC requests are sent to the bus. The bus delegates those commands to the subjects for which they were issued. The subject then executes the commands one by one.

A CNC request has three parameters:

- Subject – the object on which the command is being executed.
- Command name – the action to execute.
- Object – (optional) the object on which the command is to be executed.

All commands usually execute a predefined set of CNC requests that automatically set the subject, object (if any), command name, and command parameters based on the context from which the commands are invoked.

Because communication to the bus is asynchronous, the return from the CNC request does not guarantee that an action was successful. However, you can configure the bus to send alerts when major events happen or to respond to CNC requests. See Chapter 5, "Configuring and Using Alerts."

Note See the *e-Biz Impact Command Line Tools Guide* for more information about CNC request parameters.

Executing Global Console commands

CNC commands are available from three Global Console areas:

- All Tasks menu – right-click an object's icon in the tree view and select All Tasks to see the commands that are available for that object. The All Tasks menu option is available for these objects:

  - e-Biz Impact 5.4 Global Console node
  - Agent node
  - Cluster node
  - Controller nodes
  - AIM nodes
  - SFM nodes
  - SFM destination node
- Object right-click menu – right-click an object's icon in the tree view. Available commands are listed at the top of the pop-up menu that displays. The pop-up menu displays commands for these objects:
  - Cluster nodes
  - Controller nodes
  - AIM nodes
  - SFM nodes
  - SFM destination node
- SFM views right-click menu – some SFM and SFM destination views have a right-click menu available in the right pane that displays the view. For example, when you select the Pending Transaction view for an SFM, right-click a pending transaction in the right pane to see commands that can be executed on that transaction.

Note A user can also execute CNC requests from the command line. See the *e-Biz Impact Command Line Tools Guide* for more information.

Viewing session information

All objects that have command actions available also provide Session Info on the All Tasks menu.

When you right-click a cluster, a controller, a controller application, an SFM, or an SFM destination or source in the tree view, and select All Tasks | Session Info, you see CNC request information about the current session for the selected object. For example, if you right-click a cluster in the tree view and select All Tasks | Session Info, a window displays information similar to this:

```
Information about the current Command and Control session:
- Console is connected to cluster "MultiProcess" on host "localhost" external
port "5555"
- User "admin" is logged in
- Session is opened since "6/8/2005 8:01:06 PM"
```

Click OK to close the window.

Executing Global Console tasks

All Tasks menu

Right-click the e-Biz Impact 5.4 Global Console node in the tree view, then select All Tasks to select these commands:

Task Corresponding CNC command	Description
Import *Not applicable*	Import a file that contains a list of agents and select specific agents to import. This allows you to import agents created in a different global Console. **Note** The import function requires an input file that describes agent attributes using XML version 1.0 syntax.
Export *Not applicable*	Select all or some agents and export those agents and their attributes to a user-specified file, which can be imported into a different Global Console. The file is formatted using XML version 1.0 syntax.

Executing agent tasks

All Tasks menu

Right-click an agent's node in the tree view, then select All Tasks to select these commands:

Task Corresponding CNC command	Description
Connect connect	Connect to the e-Biz Impact server running on the host that was specified when the agent was created.

Task Corresponding CNC command	Description
Disconnect disconnect	Disconnect from the e-Biz Impact server running on the host that was specified when the agent was created.

Executing cluster tasks

All Tasks menu

To access cluster tasks, right-click the cluster's icon in the tree view and select All Tasks. The following commands are available:

Task Corresponding CNC command	Description
Reload reload	Reloads a cluster's configuration data from its configuration *.xml* file and applies any configuration changes.
Shutdown shutdown	Request a cluster to stop all of its SFM objects, controller objects, and applications managed by the controllers.
Change Password password	Changes the current operator's password. Operators can change their password at any time.
Session Info	See "Viewing session information" on page 50.

Additional cluster commands

Right-click the cluster's icon in the tree view to select these commands:

Task Corresponding CNC command	Description
Logout logout	Right-click the cluster's icon in the tree view and select Logout to log out of the e-Biz Impact server on which the cluster is running.
Login login	If you are not logged in to the e-Biz Impact server, right-click the cluster's icon in the tree view and select Login to log in to the e-Biz Impact server on which the cluster is running.

Executing controller tasks

All Tasks menu

All controller objects have session information available from the All Tasks menu. Right-click a controller node, select All Tasks, then select Session Info. See “Viewing session information” on page 50.

Additional controller commands

All controller objects allow you to enable or disable the controller’s child applications. Because a controller is a process, disabling a controller causes the controller process to terminate properly.

Task Corresponding CNC command	Description
Enable enable	Request a controller to enable the controller’s child applications.
Disable disable	Request a controller to a disable the controller’s child applications.

Executing SFM tasks

All Tasks menu

Right-click the SFM’s icon in the tree view and select All Tasks. The following commands are available:

Task Corresponding CNC command	Description
Accept acceptTransactions	Requests the SFM to accept the dispatch of transactions to all of its destinations. **Note** This command is only available after the Refuse task has been used.

Task Corresponding CNC command	Description
Refuse refuseTransactions	Requests the SFM to refuse to dispatch transactions to all of its destinations. You can set a particular SFM to refuse mode to prevent the SFM from accepting any incoming transactions. All transactions submitted to an SFM in refuse mode are returned with an error code that indicates the SFM is in refuse mode. The default is to set the SFM to Accept mode, where the SFM accepts and processes incoming transactions. This command reverses the effect of the Refuse transaction command.
Resend Transactions • pauseDestsBySerial • resendTransactionBySerial • resumeDestsBySerial	Requests the SFM to dispatch all transactions to all destinations. Completed transactions can be resent to all destinations for which they qualify. This causes the SFM to generate a new transaction, linking it to the original serial number by its progenitor. The resulting transaction follows the path of a regular transaction. Routes are defined by the qualification phase. When produced, output transactions are dispatched to all qualified destinations.
Send New Transaction sendToSFM	Send a new transaction to the SFM. Sending a new transaction allows you to test SFM internal production objects. The editor allows you to load or edit the content of a transaction, select the path to which the transaction is sent to within the SFM (engine, production object, or route), then send the new transaction. A diagnostic log indicates any results as well as the new transaction serial number when successful. **Note** This task displays the Sending New Transaction window in the right pane of the Global Console.
Pause All Destinations pauseAllDest	Requests the SFM to stop dispatching transactions to all of its destinations.
Resume All Destinations resumeAllDest	Request the SFM to resume dispatching transactions to all destinations.
Session Info	See “Viewing session information” on page 50.

Additional SFM commands

Right-click the SFM’s icon in the tree view to select these commands:

Task Corresponding CNC command	Description
Enable enable	Enable an SFM’s child applications.

Task Corresponding CNC command	Description
Disable disable	Disable an SFM's child applications.
Restart resend	Reloads the SFM's configuration data from its configuration file and restarts the SFM's controller and applications.

Executing SFM view's commands

The following commands are available from various SFM views (see "Displaying SFM application views" on page 34). When a particular SFM view has been selected, right-click in the right pane to select a command.

SFM Sources and Destinations view

When you display an SFM's sources and destinations, the following commands are available when you right-click a destination in the right pane.

Task Corresponding CNC command	Description
Pause pauseDest	Requests the SFM to stop dispatching transactions to the selected destination.
Resume resumeDest	Requests the SFM to resume dispatching transactions to the selected destination.
Repair (various)	Invokes the Repair [Unprocessable/Cancelled/Unrouteable] Transactions wizard. See "Repairing unprocessable, cancelled, and unrouteable d transactions" on page 58.
Pause All pauseAllDest	Request an SFM object to stop dispatching transactions to all of its destinations.
Resume All resumeAllDest	Request an SFM object to resume dispatching transactions to all destinations.

SFM Unrouteable Transactions view

When you display an SFM's unrouteable transactions, the following commands are available when you right-click a transaction in the right pane.

Task Corresponding CNC command	Description
Delete deleteUnrouteableTransaction	Deletes the selected transactions.

Task Corresponding CNC command	Description
Repair (various)	Invokes the Repair [Unprocessable/Cancelled/Unrouteable] Transactions wizard. See “Repairing unprocessable, cancelled, and unrouteable d transactions” on page 58.
Resend • pauseDestsBySerial • resendTransactionBySerial • resumeDestsBySerial	Requests the SFM to resend the selected transactions to all destinations. Completed transactions can be resent to all destinations for which they qualify. This causes the SFM to generate a new transaction, linking it to the original serial number by its progenitor. The resulting transaction follows the path of a regular transaction. Routes are defined by the qualification phase. When produced, output transactions are dispatched to all qualified destinations.

SFM Pending Transactions view

When you display an SFM’s sources and destinations, the following commands are available when you right-click on a destination in the right pane.

Task Corresponding CNC command	Description
Skip skipTransactionBySerial	Skip the selected transaction.
Cancel • pauseDestsBySerial • cancelTransactionBySerial • resumeDestsBySerial	Cancel the selected transactions. Cancelling transactions allows an SFM to resume normal activity and allows you to troubleshoot the problem transaction at a later time. **Note** You must select the SFM Cancelled Transactions view to uncancel a cancelled transaction.
View Data getTransactionData	View data for the selected transactions in a text editor.
Save Data getTransactionData	Saves data for the selected transactions in a text editor file.

SFM Unprocessable Transactions view

When you display an SFM’s unprocessable transactions, the following commands are available when you right-click on a transaction in the right pane.

Task Corresponding CNC command	Description
Skip skipTransactionBySerial	Skip the selected transaction.

Task Corresponding CNC command	Description
Cancel • pauseDestsBySerial • cancelTransactionBySerial • resumeDestsBySerial	Cancel the selected transactions. Cancelling transactions allows an SFM to resume normal activity and allows you to troubleshoot the problem transaction at a later time. **Note** You must select the SFM Cancelled Transactions view to uncancel a cancelled transaction.
Reprocess reprocessUnprocessableTransaction	Requests the SFM to reprocess the selected transaction.
Repair (various)	Invokes the Repair [Unprocessable/Cancelled/Unrouteable] Transactions wizard. See “Repairing unprocessable, cancelled, and unrouteable d transactions” on page 58.

SFM Cancelled Transactions view

When you display an SFM’s cancelled transactions (see “Cancelled Transactions view” on page 43), the following commands are available when you right-click on a transaction in the right pane.

Use the this view to uncancel, repair, or skip a transaction.

Task Corresponding CNC command	Description
Uncancel • pauseDestBySerial • uncancelTransactionBySerial • resumeDestBySerial	Uncancels the selected transactions.
Skip skipTransactionBySerial	Skip the selected transaction.
Repair (various)	Invokes the Repair [Unprocessable/Cancelled/Unrouteable] Transactions wizard. See “Repairing unprocessable, cancelled, and unrouteable d transactions” on page 58.

Repairing invalid destinations

Invalid destinations exist when the SFM is restarted with transactions pending for a removed or invalid destination. When a transaction is submitted to SFM, it is qualified against production objects, if needed, and stored in the log file with a PENDING status. In the log file, the destinations to which the transaction will be dispatched are marked. When the SFM is shut down and restarted, it reloads all pending transactions into memory and puts them back onto the internal queue for dispatch. However, if between the shut down and restart, destinations for pending transactions are removed from the SFM configuration, the SFM cannot dispatch the transactions because the intended destination no longer exists. Transactions that are rerouted to a valid destination are marked REPAIRED in the log file.

Repairing unprocessable, cancelled, and unrouteable d transactions

When you select Repair for an unprocessable, cancelled, or unrouteable transaction, the Repair [Unprocessable/Cancelled/Unrouteable] Transaction wizard displays, which has three tabs—Edit, Action, and Finish.

- Edit – edit the transaction data, fixing what might be causing the transaction to be unprocessable or caused the transaction to be cancelled.
- Action – identify whether you want the Global Console to reprocess the transaction for all routes or for specified routes (such as Pending). Select Show Destination Names to view specific destinations, which may be helpful if you are routing a transaction to multiple AIMs with different names.
- Finish – click Finish to exit the wizard and repair the transaction.

Note Repairing a transaction generates a new transaction and links the initial transaction with the new transaction, which retains history and preserves the original transaction's content.

Skipping transactions

This task can pull out a transaction already queued for delivery, and using the serial number, the name of the production objects for which the transaction is qualified, and the delivery destination, skip the transaction. The skipped transaction is moved to the completed log and considered complete. Skipped or completed transactions can be recycled to the SFM using the Resend or Resend All menu options.

Executing destination tasks

All Tasks menu

Right-click the destination's icon in the tree view and select All Tasks. The following commands are available:

Task Corresponding CNC command	Description
Resend Transactions • pauseDestsBySerial • resendTransactionBySerial • resumeDestsBySerial	Requests the SFM to dispatch all transactions to the selected destination.
Session Info	See "Viewing session information" on page 50.

Additional destination commands

Right-click the destination's icon in the tree view to select these commands:

Task Corresponding command	Description
Pause pauseDestsbySerial	Requests the SFM to stop dispatching transactions to the selected destination.
Resume resumeDestsbySerial	Requests the SFM to resume dispatching transactions to the selected destination.

CHAPTER 4

Monitoring e-Biz Impact Events

This chapter explains how to use the e-Biz Impact Event Monitor, which allows you to view Simple Network Management Protocol (SNMP) traps and drill down to see each event's detail.

Introduction

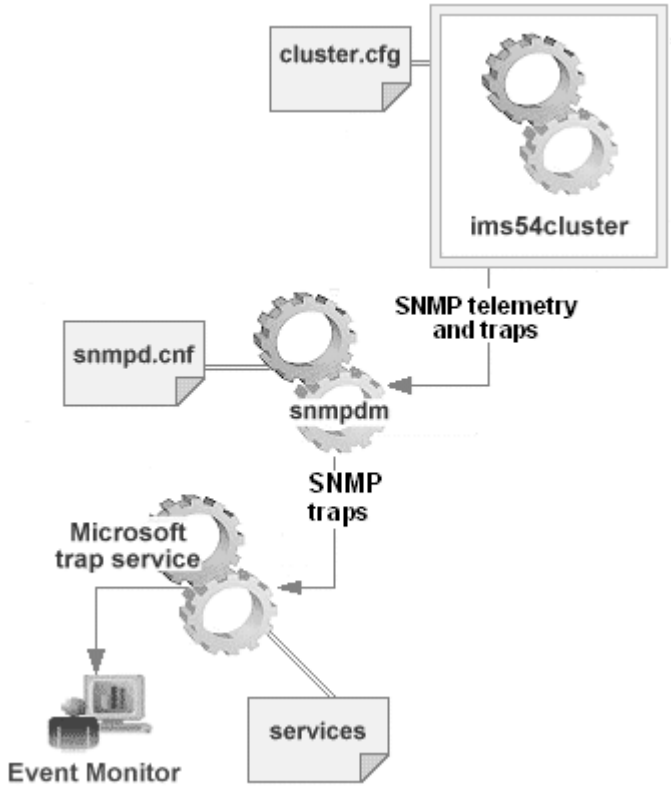
The Event Monitor—a Microsoft Management Console (MMC) snap-in—allows you to view Simple Network Management Protocol SNMP traps. SNMP traps enable an agent to notify the Event Monitor of significant events by way of an unsolicited SNMP message.

The Event Monitor displays events collected from one or more e-Biz Impact servers. These events are triggered by e-Biz Impact and provide information, warnings, or errors that occur at runtime.

Note e-Biz Impact gathers trapped event messages using the Microsoft SNMP Trap Service and publishes the messages to the Event Monitor using the SNMP EMANATE Master Agent service.

These traps are described in the e-Biz Impact standard Management Information Base (MIB) file—*impact8.my*—located in *x:\Sybase\ImpactServer-5_4\snmp* on Windows and *x/ImpactServer-5_4/snmp* on UNIX, where "*x*" is the drive or directory where the e-Biz Impact server is installed.

Figure 4-1: Event Monitor process flow



Note The Event Monitor does not listen for traps published by the Open Transport driver, or offer advanced alert management operations like command and control or display on mobile devices. Advanced alert management operations are available with the Alert Management System, which is part of the Sybase BizTracker product. See your Sybase representative for details.

The Event Monitor is installed on Microsoft Windows as part of the e-Biz Client installation, which copies the files to the *x:\Sybase\ImpactClient-5_4\EventMonitor* directory, where "*x*" is the drive where the e-Biz Impact client is installed.

Setting up event monitoring

Monitoring cluster events requires several configuration steps. Which steps are necessary depends on whether you want to publish traps to a custom port:

1 Install the Microsoft SNMP Trap Service. See “Installing the Microsoft SNMP Trap Service” on page 63.

2 Disable the SNMP Service. See “Disabling the Windows SNMP Service” on page 64.

3 Configure the SNMP EMANATE Master Agent service. This step is necessary only if you are sending traps to a custom port or multiple machines. See “Configuring the SNMP EMANATE Master Agent service” on page 64.

4 Configure e-Biz Impact clusters to publish alerts to the SNMP service. See “Configuring clusters to publish SNMP alert traps” on page 70.

When you complete the configuration, you can begin monitoring event activity on e-Biz Impact servers. See Chapter , “Monitoring events.”

Installing the Microsoft SNMP Trap Service

The e-Biz Impact Event Monitor requires the Microsoft SNMP Trap Service. By default, Windows does not have SNMP installed. Therefore, you must install this service, which is distributed by Microsoft as a component of Management and Monitoring Tools on Windows 32-Bit platforms.

❖ **Installing the Microsoft SNMP Trap Service**

1 Select Start | Settings | Control Panel | Add Remove Programs.

2 In the Add/Remove Programs window, select Add/Remove Windows Components in the left pane.

3 When the Windows Components wizard appears, select Management and Monitoring Tools in the Components list.

4 Click Details, select Simple Network Management Protocol if it is not already selected, then click OK.

5 Click Next in the Window Components wizard. The system begins configuring the selected component.

6 When the windows states that you have successfully completed the wizard, click Finish.

7 Close the Add/Remove Programs window.

Note If the installation tries to start the SNMP Service but fails, ignore the failure and continue the installation. You are going to disable the SNMP Service in the next section anyway.

Disabling the Windows SNMP Service

Because e-Biz Impact uses the SNMP EMANATE Master Agent instead of Windows SNMP agent (SNMP Service), you should disable the Windows agent. This ensures that SNMP EMANATE Master Agent service runs by default when you restart your computer:

1 Select Start Settings | Control Panel | Administrative Tools | Services.

2 In the Services window, right-click SNMP Service and select Properties.

3 In the Properties window on the General tab, for Startup Type, select Disable from the drop-down list.

4 Click OK, then close the Services window.

Configuring the SNMP EMANATE Master Agent service

This section explains how to push SNMP traps to a custom port (“Pushing traps to a custom port and machine, or multiple machines” on page 67) and how to push traps to multiple ports (“Pushing traps to a custom port and machine, or multiple machines” on page 67). This configuration must be done for each instance of the SNMP EMANATE Master Agent service (snmpdm) process.

Note If you are pushing traps to the default port 162 and only want to push traps to one port, skip this section and go to “Configuring clusters to publish SNMP alert traps” on page 70.

How the SNMP EMANATE Master Agent service works

e-Biz Impact server uses the SNMP EMANATE Master Agent service (snmpdm) to create SNMP traps and publish them to the Event Monitor. The SNMP EMANATE Master Agent service is installed with the e-Biz Impact server.

The SNMP EMANATE Master Agent knows where to publish traps based on the entries in two files—the Windows *services* file, and the SNMP EMANATE Master Agent *snmpd.cnf* configuration file.

services file

The Windows *services* file is located in *C:\WINNT\system32\drivers\etc* and specifies the port used by the Windows SNMP Service and the Windows SNMP Trap Service, as indicated by these lines in the file:

```
snmp          161/udp                     #SNMP
snmptrap      162/udp      snmp-trap      #SNMP trap
```

The first line indicates the Windows agent used for SNMP, which you can ignore. Because e-Biz Impact uses the SNMP EMANATE Master Agent instead, you disabled the Windows agent (`snmp`) running on port 161 in "Disabling the Windows SNMP Service" on page 64.

The second line lists the default port for SNMP traps. If you need to use a port other than 162 for SNMP traps, you must change the default entry in this file. See "Pushing traps to a custom port and machine, or multiple machines" on page 67.

snmpdm.cnf file

The SNMP EMANATE Master Agent *snmpd.cnf* configuration file is located in *x:\Sybase\ImpactServer-5_4\snmp\srconf\agt* on Windows and *~/Sybase/ImpactServer-5_4/snmp/srconf/agt* on UNIX (where "*x*" and "~" is the drive, file system, or directory where the e-Biz Impact Server is installed).

When the SNMP EMANATE Master Agent is installed, the installation sets the *SR_AGT_CONF_DIR* system variable, which specifies where the agent's configuration file—*snmpd.cnf*—is located.

The configuration file contains two predefined sections for trap notifications—`#Entry type: snmpNotifyEntry` (Figure 4-2) and `#Entry type:snmpTargetAddrEntry` (Figure 4-3).

- `#Entry type: snmpNotifyEntry` – contains the entries `snmpNotifyEntry 31 Console trap nonVolatile` and `snmpNotifyEntry 32 TrapSink trap nonVolatile`, which are used for trap notifications.

Figure 4-2: #Entry type: snmpNotifyEntry

```
#Entry type: snmpNotifyEntry
#Format:  snmpNotifyName  (text)
#         snmpNotifyTag  (text)
#         snmpNotifyType  (trap(1), inform(2))
#         snmpNotifyStorageType  (nonVolatile, permanent, readOnly)
snmpNotifyEntry  31 Console trap nonVolatile
snmpNotifyEntry  32 TrapSink trap nonVolatile
```

- `#Entry type:snmpTargetAddrEntry` – has the entry `snmpTargetAddrEntry 31 snmpUDPDomain 127.0.0.1:0 100 3 Console \ v1ExampleParams nonVolatile 255.255.255.255:0 2048`, used by the server to determine the target address to which traps are published.

Figure 4-3: #Entry type:snmpTargetAddrEntry

```
#Entry type: snmpTargetAddrEntry
#Format:  snmpTargetAddrName  (text)
#         snmpTargetAddrTDomain  (snmpUDPDomain, snmpIPXDomain, etc.)
#         snmpTargetAddrTAddress  (transport address,i.e.
#                                 192.147.142.254:0)
#         snmpTargetAddrTimeout  (integer)
#         snmpTargetAddrRetryCount  (integer)
#         snmpTargetAddrTagList  (text)
#         snmpTargetAddrParams  (text)
#         snmpTargetAddrStorageType  (nonVolatile, permanent, readOnly)
#         snmpTargetAddrTMask  (transport mask, i.e. 255.255.255.255:0)
#         snmpTargetAddrMMS  (integer)
snmpTargetAddrEntry  31 snmpUDPDomain 127.0.0.1:0 100 3 Console \
 v1ExampleParams nonVolatile 255.255.255.255:0 2048
snmpTargetAddrEntry  32 snmpUDPDomain 127.0.0.1:0 100 3 Console \
 v2cExampleParams nonVolatile 255.255.255.255:0 2048
snmpTargetAddrEntry  33 snmpUDPDomain 127.0.0.1:0 100 3 TrapSink \
 v3ExampleParams nonVolatile 255.255.255.255:0 2048
...
...
...
```

Note Traps are SNMP version 1 and use viExampleParams.

The default entry publishes traps to `127.0.0.1:0`. Entry "`127.0.0.1`" represents the default IP address of localhost, which it is assumed is also where the e-Biz Impact client and the Event Monitor are installed. Entry "`:0`"represents the default SNMP trap service port 162.

If the Event Monitor (and the e-Biz Impact client) are installed on a different machine, or to publish traps to multiple machines, you must create an entry for each machine's IP address. If you changed the SNMP trap port in the *services* file, you must also change the port number here. See "Pushing traps to a custom port and machine, or multiple machines" on page 67.

Pushing traps to a custom port and machine, or multiple machines

This section explains how to enter a custom port number and multiple IP addresses for SNMP trap publication.

1 On the Windows machine where the e-Biz Impact client is installed, go to *C:\WINNT\system32\drivers\etc* and open the *services* file in a text editor.

2 Find this line:

```
snmptrap      162/udp      snmp-trap      #SNMP trap
```

and change "`162`" to the number of the custom port used for the trap service.

3 Save the *services* file and close the text editor.

4 Restart the Windows machine.

5 Temporarily stop the SNMP EMANATE Master Agent service:

On Windows:

a Select Start | Settings | Control Panel | Administrative Tasks | Services.

b In the Services window, find the SNMP EMANATE Master Agent service in the right pane.

c Right-click the service and select Stop. Leave the Services window open but minimized.

On UNIX:

a Verify that the snmpdm process is stopped. In a terminal window, go to the directory where the e-Biz Impact server is installed and enter the following command:

```
$ps
```

You should see a list of processes output to the screen similar to this:

```
PID     TTY       TIME     COMMAND
8347    rt021a0   0:03     ksh
8376    rt021a0   0:06     ps
```

b If any process beginning with "snmpdm" is running, write down the process identification number (PID) for that process, then kill the process:

```
$kill -2 PID
```

where *PID* is the process identification number of the snmpdm process.

6 Create a backup copy of the *snmpd.cnf* configuration file, which is located in *x:\Sybase\ImpactServer-5_4\snmp\srconf\agt* on Windows and in *~/Sybase/ImpactServer-5_4/snmp/srconf/agt* on UNIX (where "*x*" and "~" is the directory or system on which the e-Biz Impact Server is installed)

7 Open *snmpd.cnf* in a text editor.

Note Because of default file properties, on Windows the *snmpd.cnf* file may display as a shortcut, may have a description of Speed Dial, and may not display an extension. To open the file in a text editor, right-click the file and select Open With, select a text editor, then click OK.

8 To specify a custom port number:

a Go to the section that begins with:

```
#Entry type: snmpTargetAddrEntry
```

b In this section only, find each occurrence of:

```
snmpUDPDomain 127.0.0.1:0
```

c Change each entry you find to:

```
snmpUDPDomain 127.0.0.1:<custom_port_number>
```

where <*custom_port_number*> is the custom port number you entered in the *services* file in step 3.

9 To publish traps to a different IP address:

a Go to the section that begins with:

```
#Entry type: snmpTargetAddrEntry
```

b In this section only, find each occurrence of:

```
snmpUDPDomain 127.0.0.1:0
```

c Change each entry you find to:

```
snmpUDPDomain <Event_Monitor_host_IP>:0
```

where <*Event_Monitor_host_IP*> is the IP address of the host that runs the Event Monitor.

Warning! Do not use the machine name or "localhost." You must use the IP address.

---

10 To publish traps to multiple machines:

a Go to the section that begins with:

```
#Entry type: snmpTargetAddrEntry
```

b Copy entry 31:

```
snmpTargetAddrEntry  31 snmpUDPDomain 127.0.0.1:0 100 3 Console \
     v1ExampleParams nonVolatile 255.255.255.255:0 2048
```

c Paste the copied entry above entry 31.

d Change "`31`" to "`30`" and change `127.0.0.1` to the additional IP address.

e Repeat this process and create a new entry (`29`, `28`, and so on) for each additional machine on which you have an Event Monitor installed.

f Go to the section that begins with:

```
#Entry type: snmpNotifyEntry
```

g Create an entry for each new entry you created in steps "a" through "e." For example, if you created new entries 30, 29, and 28, you would create these entries in the `#Entry type: snmpNotifyEntry` section above entry 31:

```
snmpNotifyEntry 28 Console trap nonVolatile
snmpNotifyEntry 29 Console trap nonVolatile
```

```
snmpNotifyEntry 30 Console trap nonVolatile
```

Warning! Creating too many new entries can have a significant impact on e-Biz Impact performance.

11 Save the *snmpd.cnf* file and close the text editor.

12 Restart the SNMP EMANATE Master Agent service on Windows or UNIX.

Configuring clusters to publish SNMP alert traps

To enable clusters to publish alerts to the SNMP service:

1 Select Start | Programs | Sybase |e-Biz Impact 5.4 | Configurator.

2 When the Configurator window opens, right-click the e-Biz Impact Configurator node in the tree view and select Load Cluster.

3 When the Open dialog box appears, navigate to the cluster's *.xml* file, select it, then click Open.

4 Right-click the cluster in the tree view and select Properties.

5 When the cluster properties window opens, select the General tab, select the Publish Alerts to SNMP Service option in the Monitoring section, then click OK.

6 Right-click the cluster in the tree view and select Save to save the cluster with the same name.

7 Repeat steps 2 through 6 for each cluster that you want to monitor events for in the Event Monitor.

8 Select Console | Exit to close the Configurator.

Monitoring events

This section explains how to use the Event Monitor to see informational, warning, or events for your e-Biz Impact clusters.

Note The Event Monitor runs only on Windows systems.

❖ **Starting the Event Monitor**

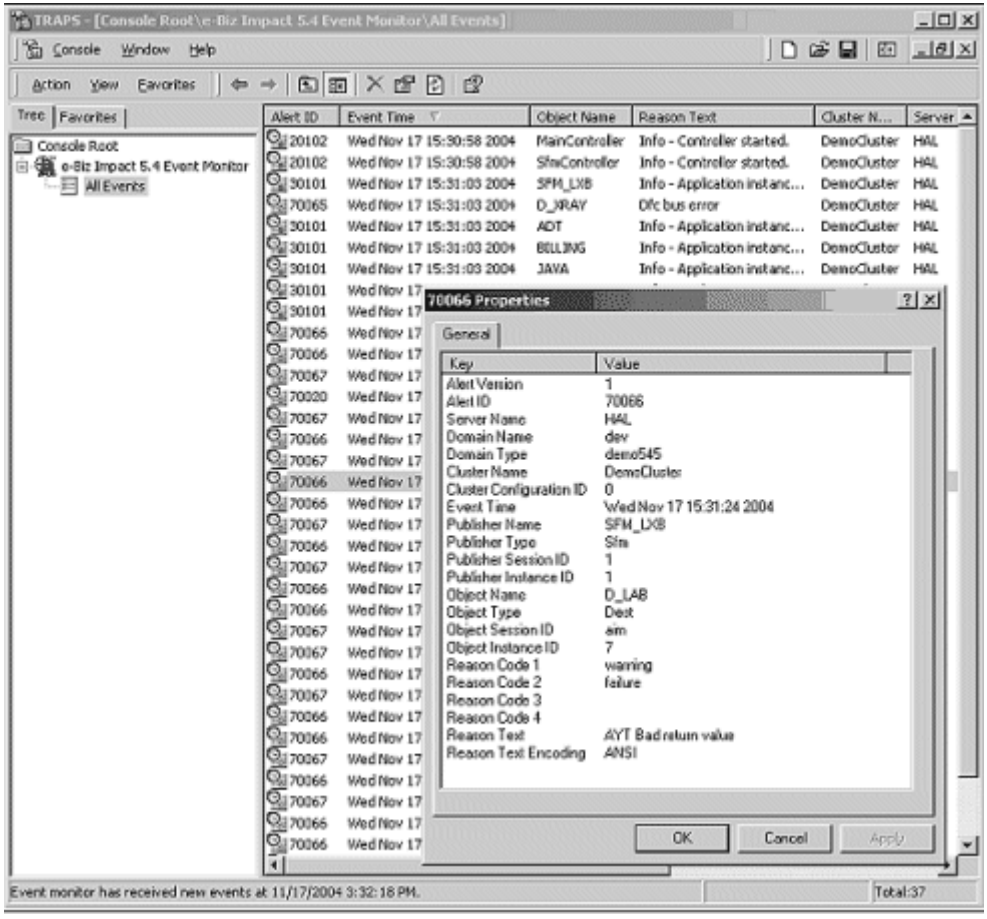
1 Select Start | Programs | Sybase | e-Biz Impact 5.4 | Event Monitor.

2 When the Event Monitor console displays, right-click e-Biz Impact 5.4 Event Monitor in the tree view, and select Start.

This action starts the receipt of SNMP traps and allows you to create an optional filter to display all received events received. The console checks to see if the SNMP trap service is missing or disabled.

3 Select the All Events node.

Figure 4-4: Event Monitor



Events display in the right pane.

4 Double-click an event to see its entire content.

5 You can select one or more events and delete events.

Changing Event Monitor properties

The Event Monitor keeps a collection of all events received. To minimize the number of events stored in this collection, specify which events to collect based on predefined event severity (info, errors, warnings) in the Event Monitor's properties window.

Right-click e-Biz Impact 5.4 Event Monitor in the tree view and select Properties. When the Properties window displays, on the General tab select the severity of events for which you want to receive information—Information, Warnings, and Errors. You can select all three options if you choose.

The Information tab displays version and file information for your reference.

When you finish, click Apply, then click OK.

CHAPTER 5

Configuring and Using Alerts

This chapter describes how to configure e-Biz Impact support for Open Transport-XML (OT-XML) alerts.

Introduction

e-Biz Impact alerts provide developers with a way to respond programmatically to predefined cluster activities that require user intervention. e-Biz Impact alerts allow a developer or administrator to:

- Publish Open Transport-XML alerts when a specified event occurs on a monitored system or subsystem.
- Read alert messages from a queue or file device and invoke a user-maintained shell script or binary to act on the alert.

You configure alerts using sample configuration files packaged with e-Biz Impact and available at installation. The samples are located in *x:\Sybase\ImpactServer-5_4\samples\alerts* on Windows and *~/Sybase/ImpactServer-5_4/samples/alerts* on UNIX, where "*x*" and "~" are the drive, file system, and directory where the e-Biz Impact server is installed.

You configure alerts using sample configuration files packaged with e-Biz Impact and available at installation. The samples are located in *x:\Sybase\ImpactServer-5_4\samples\alerts* on Windows and *~/Sybase/ImpactServer-5_4/samples/alerts* on UNIX, where "*x*" and "~" are the drive, file system, and directory where the e-Biz Impact server is installed.

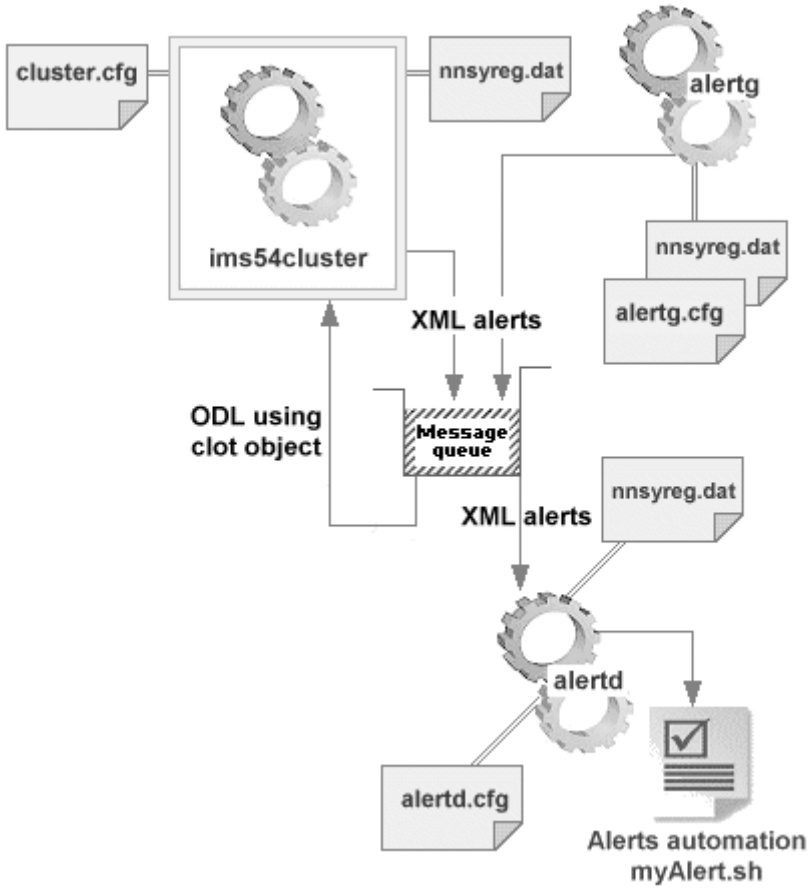
Note When you enter path names, remember that names are not case sensitive on Windows, but are case sensitive on UNIX systems.

---

e-Biz Impact support for Open Transport-XML alerts uses these executables:

- ims54alertg (alertg), and ims54cluster (UNIX, Windows) – publishes OT-XML alerts. To use the alert information from ims54cluster or ims54alertg, an OT-XML message on the alert transport, such as an item in the PENDING queue, must be turned into a user-defined action using alertd.
- ims54alertd(alertd) (UNIX, Windows) – reads alert messages from the alert transport, such as a queue or file device, and invokes a user-maintained shell script or binary.

Figure 5-1: e-Biz Impact alerts overview



Both ims54cluster and alertg can publish OT-XML alerts. After an alert is published, an external handler is required to manage the alert, and takes some user-defined action such as issuing e-mail or logging the event.

You specify the values for your transports in the following sample files:

- Use the *alertg.cfg* file to specify context and transports for alertg.
- Use the *alertd.cfg* file to specify context and transports for alertd.
- Use the *nnsyreg.dat* file to configure context and transports for multiple applications, such as alertg and alertd.

Configuring OT-XML alerts

To publish OT-XML alerts, you must perform these configuration steps:

- Set the *PATH* environment variable for the *ims* wrapper script, which is used to execute the alert operations. See “Setting the ims wrapper script path” on page 79.
- Verify the *ICU_DATA* environment variable setting. This variable must be set to run the alerts service as a Windows service or UNIX daemon. See “Verifying the ICU_DATA variable setting” on page 81.
- Use the e-Biz Impact Configurator to set up your clusters to send alerts as XML to Open Transport. See “Configuring clusters to send XML alerts” on page 82 for instructions.
- Configure an alertg configuration file. See “Configuring alertg to publish XML alerts” on page 83.
- Configure an alertd configuration file. See “Configuring alertd for OT-XML alerts” on page 87.
- Configure alerts to run as a Windows service or UNIX daemon (optional). See “Configuring alertd to run as a service or daemon” on page 95.
- Configure an *nnyreg.dat* file. See “Configuring the OT nnsyreg.dat configuration file” on page 90.

Note When you enter path names, remember that names are not case sensitive on Windows, but are case sensitive on UNIX systems.

Guidelines for editing alert configuration files

- Left-align a configuration stanza. This line serves as a grouping header and is not indented or commented. Subsequent lines with keys in the form *XXX = YYY* are members of the stanza. The fully qualified name of a key named *XXX* within a stanza named is *AAA* is of the form: *AAA.XXX*.
- Indent all configuration keys with spaces or tabs. Use the form *XXX = YYY*, where *XXX* is the key name and *YYY* is the key value.
- Begin a comment line with a pound character (#).

Setting the *ims* wrapper script path

The ims wrapper script is used to execute alert commands. The *PATH* environment variable is used by the system to find the ims command within the directory hierarchy. If you execute an ims command in a terminal window (UNIX systems) or at a command line (Windows) and do not have the *PATH* variable set correctly, the system cannot find the command unless you include the full path to the executable when you type the command.

The ims executable (*ims.cmd*) is located in *x:\Sybase\ImpactServer-5_4\bin* on Windows and in *~/Sybase/ImpactServer-5_4/bin* on UNIX systems (where "*x*" and "~" represent the location where the e-Biz Impact 5.4.5 server is installed).

❖ **Setting the *PATH* variable for the *ims* script on Windows**

1 Select Start | Settings, then double-click Control Panel.

2 When the Control Panel window opens, double-click the System icon.

3 When the System Properties window opens, select the Advanced tab, then click Environment Variables.

4 In the System Variables section of the window, select the *PATH* variable and click Edit.

5 In the Edit System Variable dialog box, append the ims wrapper script location to the *PATH* variable. For example, if the e-Biz Impact server is installed in the root of drive D: on Windows, enter this value at the end of the *PATH* variable in the Variable Value field:

```
;D:\Sybase\ImpactServer-5_4\bin
```

Note Separate the existing path statement and the new entry with a semicolon (;) with no space after the semi-colon.

6 Click OK to save your entry and close the Edit System Variable dialog box.

7 Click OK to close the Environment Variable window, then click OK to close the System Properties window.

8 Close the Control Panel window.

9 Restart your machine to implement the modified *PATH* variable.

❖ **Setting the *PATH* variable for the *ims* script on UNIX systems**

This procedure is for the C-shell user environment. If your UNIX system uses a different shell (for example, the Bourne shell or Korn shell), see your UNIX system documentation for instructions on setting the *PATH* variable.

Note To see which shell is the default on your system, type `echo $SHELL` in a terminal window.

You can set the *PATH* variable each time you open a terminal session and start the cluster, or you can set the *PATH* variable permanently for all terminal sessions by adding the ims wrapper script path to the *$HOME/.cshrc* file.

1 To permanently add the ims executable to your path:

a Open the *$HOME/.cshrc* file in a text editor.

b Add the following line to the *.cshrc* after the list of other commands:

```
set path = ($path /~/Sybase/ImpactServer-5_4/bin)
```

where "~" is where the e-Biz Impact server is installed.

c Save the *.cshrc* file and close the text editor.

d Log out of the system and log back in to establish the new path.

2 To add the ims path to the end of your existing path for only the current session, open a terminal window and enter the following command on one line:

```
set path = ($path /~/Sybase/ImpactServer-5_4/bin)
```

where "~" is where the e-Biz Impact server is installed.

Note If you did not use step 1 to permanently add the ims wrapper script location to your path, each time you open a terminal window to execute the cluster, you must first enter the command shown in step 2 to set the *PATH* variable for the ims wrapper script for that session.

Verifying the *ICU_DATA* variable setting

To run alerts as a Windows service or UNIX daemon, the e-Biz Impact *ICU_DATA* system variable must be set. Although this variable is usually set during installation, you should verify that it exists and is set properly.

❖ **Verifying the ICU_DATA variable on Windows**

1 Select Start | Settings, then double-click Control Panel.

2 When the Control Panel window opens, double-click the System icon.

3 When the System Properties window opens, select the Advanced tab, then click Environment Variables.

4 In the System Variables section of the window, look for the *ICU_DATA* variable.

If the ICU_DATA variable is there, verify that the value is ***x***:\Sybase\ImpactServer-5_4\share\icu\data, where "*x*" is the drive and directory where the e-Biz Impact server is installed.

If the ICU_DATA variable is not there, click New below the System Variables section of the Environment Variables window. In the New System Variable dialog box, enter ICU_DATA in the Variable Name field, and enter ***x***:\Sybase\ImpactServer-5_4\share\icu\data in the Variable Value field, where "*x*" is the drive and directory where the e-Biz Impact server is installed.

Click OK.

5 Click OK to close the Environment Variable window, then click OK to close the System Properties window.

6 Close the Control Panel window.

7 If you had to add or change the *ICU_DATA* variable, restart the machine.

❖ **Verifying the ICU_DATA variable on UNIX systems**

This procedure is for the C-shell user environment. If your UNIX system uses a different shell (for example, the Bourne shell or Korn shell), see your UNIX system documentation for instructions on setting the *ICU_DATA* variable.

Note To see which shell is the default on your system, type `echo $SHELL` in a terminal window.

You can set the *ICU_DATA* variable each time you open a terminal session and start the cluster, or you can set the *ICU_DATA* variable permanently for all terminal sessions by adding the ims wrapper script path to the *$HOME/.cshrc* file.

1 To permanently add the ims executable to your path:

   a Open the *$HOME/.cshrc* file in a text editor.

   b Add the following line to the *.cshrc* after the list of other commands:

```
setenv ICUDATA /~/Sybase/ImpactServer-5_4/share/icu/data)
```

   where "~" is where the e-Biz Impact server is installed.

   c Save the *.cshrc* file and close the text editor.

   d Log out of the system and log back in to establish the new path.

2 To set the *ICU_DATA* variable for only the current session, open a terminal window and enter the following command on one line:

```
setenv ICUDATA /~/Sybase/ImpactServer-5_4/share/icu/data
```

   where "~" is where the e-Biz Impact server is installed.

Configuring clusters to send XML alerts

To enable clusters to send alerts as XML to Open Transport:

1 Select Start | Programs | Sybase |e-Biz Impact 5.4 | Configurator.

2 When the Configurator window opens, right-click the e-Biz Impact Configurator node in the tree view and select Load Cluster.

3 When the Open dialog box appears, navigate to the cluster you want to configure, select it, then click Open.

4 Right-click the cluster in the tree view and select Properties.

5 When the cluster properties window opens, select the General tab and select these options in the Monitoring section:

- Send alerts as XML to Open Transport – publish generated alerts via a configured Open Transport context name and transport name.
  - Context Name – the Open Transport context stanza name, which must match the context name in the *nnsyreg.dat* file. The name must begin with a letter, and can contain letters, numbers, and the underscore (_) character. The default name is `OTI_Context`.
  - Transport Name – identifies the Open Transport transport stanza name, which must match the transport name in the *nnsyreg.dat* file. The default is `AlertsIn`.

  See "Configuring alertg to publish XML alerts" on page 83 for more information.

6 Click OK to save your entries and close the window.

7 Right-click the cluster in the tree view and select Save to save the cluster with the same name.

8 Select Console | Exit to close the Configurator.

Configuring *alertg* to publish XML alerts

The alertg application publishes OT-XML alerts and pushes them to a message queue. This functionality requires an alertg configuration file. e-Biz Impact provides a sample alertg configuration file (*alertg.cfg*) for reference, which you can use as a template to create a customize alertg configuration file.

1 Open the sample file *x:\Sybase\ImpactServer-5_4\samples\alerts\alertg.cfg* on Windows and *~/Sybase/ImpactServer-5_4/samples/alerts/alertg.cfg* on UNIX, where "*x*" and "~" represent the drive, system, or directory where the e-Biz Impact server is installed.

Use the “Sample alertg configuration file” on page 84 for reference.

Note The “keys” in *alertg.cfg* are called “alert data” and represent the essential information published in OT-XML alerts. The alert data values for the various alerts generated by various e-Biz Impact entities, for example, the SFM, the cluster, and the controller, are compiled in the alert definition reference.

2 In the first key, PubType, specify the notification type. Choose Alert.

Warning! The PubTypes Trap and Both are not currently supported.

3 Specify the `ContextName` (the default value is `OTI_Context`) and `TransportName` (the default value is `AlertsIn`).

The value for `ContextName` must correspond to a stanza in the *nnsyreg.dat* file that uses the form `OTContext.`***`<ContextName>`***.

For this example, the stanza `OTContext.OTI_Context` must exist locally or externally. For more information, see “Configuring alertd for OT-XML alerts” on page 87.

The value for `TransportName` must also correspond to a stanza in the *nnsyreg.dat* file that uses the form `Transport.<`***`TransportName`***`>`; for example, `Transport.AlertsIn`.

4 Specify the remainder of the keys to configure information to publish as OT-XML alerts.

Sample *alertg* configuration file

The following text is the sample *alertg.cfg* configuration file used by alertg. The first part of the file—each line that is preceded by a pound sign (#)—provides comments that describe each configuration key.

The second part of the file (lines not preceded by #), contains the actual configuration data.

```
#
# Sample alertg.cfg for ims54alertg.  Keys are...

# PubType
#     Use: Defines notification type to publish
#     Allowed Values: Alert
```

```
#     Default: none
#
# ContextName
#     Use: Used for Alerts only.
#          Identifies OT context stanza name.
#     Default: OTI_Context
#
# TransportName
#     Use: Used for Alerts only. Identifies OT transport
#          stanza name.
#     Default: AlertsIn
#
# Hostname
#     Use: Sets host/server name
#     Default:
$
# DomainName
#     Use:
#     Default:
#
# DomainType
#     Use:
#     Default:
#
# ClusterName
#     Use:
#     Default:
#
# ClusterCfgID
#     Use:
#     Default: 0
#
# SourceAppl
#     Use: Used for Alerts only. PersonalAlerts.
#          PersonalAlert.SourceApplication
#     Default: Impact5.4
#
# EventID
#     Use: Sets the alert identifier
#     Allowed Values: 0..4294967295
#     Default: none
#
# Publisher*
#     Use: Who is publishing alert/trap
#     Default: none
#
```

```
# Publisher*
#     Use: What the affected entity/object is
#     Default: none
#
# Reason*
#     Use: Codes/descriptions of problem
#     Default: none
#
#########################################################

     PubType = Alert
     ContextName = OTI_Context
     TransportName = AlertsIn
     Hostname = myhost
     DomainName = myImpact
     DomainType = myTest
     ClusterName = myCluster
     ClusterCfgID = 999
     SourceAppl = Impact5.4
     EventID = 1234
     PublisherName = foobar
     PublisherType = sfm
     PublisherSessionID = 1234
     PublisherInstance = 2
     ObjectName = foobar
     ObjectType = sfm
     ObjectSessionID = 3
     ObjectInstance = 4
     ReasonCode1 = Refuse mode
     ReasonCode2 = Console
     ReasonCode3 = Fred
     ReasonCode4 =
     ReasonText = SFM 'foobar' was put in Refuse mode
     ReasonEncoding = ANSI# Open Transport (OT)
#
# Configuration can occur here or in nnsyreg.dat in the
# working directory or in the directory specified by the
# NNSY_ROOT environment variable. See sample OT
# configuration files nnsyreg_mqs.dat and
# nnsyerg_file.dat, for example, configurations for
# MQ Series and the OT file driver respectively.
```

Configuring *alertd* for OT-XML alerts

While the configuration of ims54cluster is different from alertg, an alert message from the cluster program is a structurally equivalent message on a transport device, such as the PENDING queue, and becomes the input for alertd.

Typically, alertd uses the same external OT configuration file as alertg. The alertd configuration specifies the parameters to read a message off a transport and invoke a shell script or binary.

The `Transport.Alerts` stanza identifies the related properties for the PENDING queue. In the alertg sample file, the key `TransportName` has the value `AlertsIn`, and that Transport.AlertsIn specifies the PENDING queue. The net effect of running alertg with the combined sample configurations is to place an alert message on the queue named PENDING. The other queues HISTORY and ERROR are used only by alertd.

❖ **Configuring transports**

1 Locate and open a sample file. Go to:

`$`*`NNSY_ROOT`*`/samples/alerts`

where *NNSY_ROOT* is the environment variable for the install location.

2 In the `In` stanza, specify the alert message input transport. This name appends to Transport session entry; for example, `Transport.AlertsIn`.

3 In the `PollRate` stanza, specify how long to pause before checking the transport after the transport is emptied.

The value is in seconds for both Windows and UNIX.

4 (Optional) Specify where to send messages to another queue after processing. For example, you can specify that messages to go to the HISTORY queue, or send failed messages to an ERROR queue.

If this key is not set, then processed alerts are not saved.

5 In the `Command` stanza, specify the script or binary name. This can be simply a text name or a fully qualified path.

Note Relative paths are not supported.

6 If the `Command` specified is a shell script, you must set `IsScript` to `TRUE`. For binaries, set it to `FALSE`.

7 In the `ExecDir` stanza, specify the working directory for the script/binary.

8 (For Windows) In the `Visible` stanza, set to TRUE for the script to launch in a visible command window.

This useful for debugging scripts.

Note Processing of the next alert does not begin until execution of the previous script or binary is complete.

In general, the script or binary should not block on stdin (for example, expect input from a user). The only time it makes sense to use an interactive script or binary is when running in foreground mode, for example, not as a Windows service or UNIX daemon.

Sample *alertd* configuration file

The following sample is the alertd configuration file (*alertd.cfg*) supplied with e-Biz Impact and is located in *x:\Sybase\ImpactServer-5_4\samples\alerts* on Windows and in *~/Sybase/ImpactServer-5_4/Samples/alerts* on UNIX, where "*x*" and "~" represent the install location of the e-Biz Impact server.

```
#
# Sample configuration file for alert handler
#   ims54alertd (and ims54alertdsvc on windows).

#########################################################

# The "Alerts" stanza is specific to ims54alertd.
#
# Default below indicates default values built into
#application (e.g. key can be omitted in cfg).
#
# Verbose
#     Use: Controls how much info is sent to
#           console/screen. FALSE=min, TRUE=max
#     Default: FALSE
#
# Context
#     Use: References an OT context stanza ID (see
#          "OTContext.OTI_Context" stanza below).
#     Default: OTI_Context
#
# In
#     Use: References the OT "input" transport stanza
#          ID (see "Transport.AlertsIn" stanza below).
#          This defines the alert message input
```

```
#          transport.
#     Default: AlertsIn
#
# PollRate
#     Use: Time (in milliseconds) to pause between read
#          attempts again the "In" transport.
#          Not all OT drivers support blocking timeouts
#          (e.g. file driver does not, but IBM Websphere
#          MQ (formerly MQSeries) does, etc). If
#          blocking is supported, it is by far preferred
#          to polling. Use 0 to indicate blocking. If
#          blocking does not work, use a "reasonable"
#          polling interval such as 5 seconds.
#     Default: INFINITE
#
# Log
#     Use: References the OT "log" transport stanza ID
#          (see "Transport.AlertLog" stanza below)
#          where processed alerts msgs are logged.
#          Usage is optional - if Log is not specified
#          then processed alerts are not saved.
#     Default: none
#
# Error
#     Use: References the OT "error" transport stanza
#          ID (see "Transport.AlertError" stanza
#          below). Defines transport where
#          unprocessable alerts msgs are sent.
#     Default: AlertError
#
# Command
#     Use: The name of script/binary. Can include path.
#     Default: myalert
#
# IsScript
#     Use: Indicates whether Command is a script or
#          binary. If TRUE, script must be processable
#          by cmd.exe (Windows) or /usr/bin/sh (Bourne
#          shell) on UNIX
#     Default: TRUE
#
# ExecDir
#     Use: Sets the working directory for the
#          script/binary.
#     Default: .
#
```

```
# Visible (Windows Only)
#     Use: Spawn a new console window to run
#          script/binary in.
#     Default: FALSE
#
#########################################################
Alerts
     Verbose = FALSE
     Context = OTI_Context
     In = AlertsIn
     PollRate = 3
     Log = AlertLog
     Error = AlertError
     Command = myalert
     IsScript = TRUE
     ExecDir = .
     Visible = TRUE

# Open Transport (OT) configuration can occur here or
# in nnsyreg.dat in the working directory or in the
# directory specified by the NNSY_ROOT environment
# variable. See sample OT configuration files
# nnsyreg_mqs.dat and nnsyreg_file.dat
# for example configurations for Websphere MQ and the
# OT file driver respectively.
```

Configuring the OT *nnsyreg.dat* configuration file

Open Transport is the queue mechanism used by alertg, ims54cluster, and alertd to exchange alerts as XML messages. You must configure *nnsyreg.dat*—the Open Transport configuration file—to specify which OT driver to use.

Note For information on configuring other e-Biz Impact-supported transports, such as the Java Message Service, Microsoft MSMQ, and TIBCO-Rendezvous, see the Open Transport documentation included on the e-Biz Impact 5.4.5 Sybooks CD.

As previously discussed in this chapter, OT-XML configuration keys can be located in the alertg or alertd configuration files, but placing the OT configuration into the *nnsyreg.dat* files allows the configuration to be shared by multiple applications.

Sample *nnsyreg.dat_mqs* file

After creating an IBM WebSphere MQ queue manager named QMALERTS and queues under that manager named PENDING, HISTORY, and ERROR, you can use the sample *ims54alertg* and *ims54alertd* configuration files already discussed in this chapter or the following OT configuration file to configure your transport.

The following code is a sample of the external OT configuration file *nnsyreg_msq.dat*, which is intended for use with IBM WebSphere MQ.

```
# File:    nnsyreg_mqs.dat
# Purpose: Sample Open Transport (OT) config for OT
# MQ Series driver
# Usage indicated above each stanza.
# Copy/append to nnsyreg.dat in working dir or
# dir specified by NNSY_ROOT environment variable

# Context for transports
     OTContext.OTI_Context
     NNOT_CTX_EMULATE_TM = TRUE
     NNOT_CTX_ENFORCE_TX = TRUE
     NNOT_CTX_DEFAULT_TIL_ID = DefaultTransport

# Identifies the MQS series queue manager by name
(QMALERTS)
Session.OTI_Queuing
     NNOT_SHARED_LIBRARY= dbt26mqs
     NNOT_FACTORY_FUNCTION= NNMQSSessionFactory
     NNMQS_SES_OPEN_QMGR= QMALERTS

# Default tranport information
Transport.DefaultTransport
     NNOT_FACTORY_FUNCTION = NNMQSQueueFactory
     NNOT_SHARED_LIBRARY = dbt26mqs
     NNOT_TIL_OPEN_SESSION_ID = OTI_Queuing

# The alert message input transport. Queue name PENDING
Transport.AlertsIn
     NNOT_FACTORY_FUNCTION= NNMQSQueueFactory
     NNOT_SHARED_LIBRARY = dbt26mqs
     NNOT_TIL_OPEN_SESSION_ID = OTI_Queuing
     NNOT_TIL_OPEN_TSI = PENDING

# The optional alert message history log. Queue name
HISTORY
Transport.AlertLog
     NNOT_FACTORY_FUNCTION = NNMQSQueueFactory
     NNOT_SHARED_LIBRARY = dbt26mqs
```

```
        NNOT_TIL_OPEN_SESSION_ID = OTI_Queuing
        NNOT_TIL_OPEN_TSI = HISTORY

# The place where malformed alert messages are sent.
Queue name ERROR
Transport.AlertError
        NNOT_FACTORY_FUNCTION = NNMQSQueueFactory
        NNOT_SHARED_LIBRARY = dbt26mqs
        NNOT_TIL_OPEN_SESSION_ID = OTI_Queuing
        NNOT_TIL_OPEN_TSI = ERROR
```

Note To use this configuration as is, create an IBM WebSphere MQ messaging queue manager named QMALERTS, and create queues under that manager named PENDING, HISTORY, and ERROR.

The `Transport.Alerts` stanza identifies the related properties for the PENDING queue. In the alertg configuration file, the key `TransportName` has the value `AlertsIn`, which specifies the PENDING queue. The net effect of running alertg with the combined sample configurations is to place an alert message in the queue named PENDING. The HISTORY and ERROR queues are used only by alertd.

Sample *nnsyreg_file.dat* using a file driver

The Open Transport file driver uses files that act in the same manner as queues. To configure this transport driver, you name the files *.ALERTS_PENDING*, *.ALERTS_HISTORY*, and *.ALERTS_ERROR* in the OT configuration file and the file driver automatically creates the files as needed in the working directory.

Additionally, files of the same name but with a *.ctl* suffix are created in the working directory. The *.ctl* files are control files that store the current read and write positions for the data files.

Note Do not edit any of the files created by the Open Transport File driver. Only delete the files in pairs when no application is using them. If needed, the NNGetmsg utility can be used to drain messages from the data files.

See the section "Wrapper scripts" in Chapter 1, "e-Biz Impact Command Utilities," of *e-Biz Impact Command Line Tools Guide*, and the *New Era of Networks Adapter for SAP R/3 3.9 User's Guide* for more information about using the NN-related commands.

You must specify the full path for all files referenced by the file driver. This ensures that all alerts are generated into the proper alert file.

```
# File:    nnsyreg_file.dat
# Purpose: Sample Open Transport (OT) config for
# OT file driver
# Usage indicated above each stanza.
# Copy/append to nnsyreg.dat in working dir or
# dir specified by NNSY_ROOT environment variable

# Context for transports
OTContext.OTI_Context
     NNOT_CTX_EMULATE_TM = TRUE
     NNOT_CTX_ENFORCE_TX = FALSE
     NNOT_CTX_DEFAULT_TIL_ID = DefaultTransport

# Identifies a pseudo "queue manager"
Session.OTI_Queuing
     NNOT_SHARED_LIBRARY = dbt26file
     NNOT_FACTORY_FUNCTION = NNSesFileFactory

# Default tranport information
Transport.DefaultTransport
     NNOT_FACTORY_FUNCTION = NNMQQueueFileFactory
     NNOT_SHARED_LIBRARY = dbt26file
     NNOT_TIL_OPEN_SESSION_ID = OTI_Queuing

# The alert message input transport
# File driver will create/use files .ALERTS_PENDING
# and .ALERTS_PENDING.ctl in working dir. Transport.AlertsIn
     NNOT_FACTORY_FUNCTION = NNMQQueueFileFactory
     NNOT_SHARED_LIBRARY = dbt26file
     NNOT_TIL_OPEN_SESSION_ID = OTI_Queuing
     NNOT_TIL_OPEN_TSI = .ALERTS_PENDING
```

```
# The optional alert message history log.
# File driver will create/use files ..ALERTS_HISTORY and
# .ALERTS_HISTORY.ctl in working dir.
Transport.AlertLog
       NNOT_FACTORY_FUNCTION = NNMQQueueFileFactory
       NNOT_SHARED_LIBRARY = dbt26file
       NNOT_TIL_OPEN_SESSION_ID= OTI_Queuing
       NNOT_TIL_OPEN_TSI= .ALERTS_HISTORY

# The place where malformed alert messages are sent
# File driver will create/use files .ALERTS_ERROR and
# .ALERTS_ERROR.ctl in working dir.
Transport.AlertError
     NNOT_FACTORY_FUNCTION = NNMQQueueFileFactory
     NNOT_SHARED_LIBRARY = dbt26file
     NNOT_TIL_OPEN_SESSION_ID = OTI_Queuing
     NNOT_TIL_OPEN_TSI = .ALERTS_ERROR
```

Alert configuration values precedence

OT-XML configuration keys can be located in a alertg configuration file (or a ims54cluster configuration file), placed in the OT *nnsyreg.dat* configuration file, or entered on a Windows command line or in a UNIX terminal window.

The following list identifies the order of precedence for OT-XML configuration values.

1 Command line.

Configuration information entered at a Windows command line or in a UNIX terminal window takes precedence over configuration data located in the alertg configuration file, or in the OT configuration file (*nnsyreg.dat*).

To specify a configuration key on the command line to overwrite default configuration file values, where the key is within a stanza, use a hyphen followed by the stanza name, followed by a period, followed by the key name, followed by either an equal sign or a space, and finally the key value. For example, the follow configuration value would be entered on one line:

```
-EventID=1234, -EventID 1234,
     -Transport.AlertsIn.NNOT_TIL_OPEN_TSI=PENDING
```

2 alertg configuration file (for example, *alertg.cfg*).

For ims54cluster, the equivalent file is *<clusterName>.alert*, where *<clusterName>* is the name of the cluster.

3 *nnsyreg.dat* file in the current working directory.

4 *nnsyreg.dat* file in the directory referenced by the environment variable *NNSY_ROOT*.

Configuring *alertd* to run as a service or daemon

You can alertd as both a Windows service and a UNIX daemon.

Note In the alertd configuration file, set Alerts.Visible to FALSE before running alertd as a service or daemon.

Running alerts as a Windows service

Description

Use ims54alertdsvc (alertdsvc) to run alerts as a Windows service. The service uses extra parameters to install, set up, start, stop, restart, and remove the service.

Note You must have set the *PATH* environment variable for the ims wrapper script for these commands to execute correctly. See “Setting the ims wrapper script path” on page 79.

Syntax

ims alertdsvc {-install | -setup} *service_name* -home *dir* -file *cfg*
-start *service_name*
-stop *service_name*
-restart *service_name*
-remove *service_name*

Parameters

- -install *service_name* – installs alerts as a Windows service, where *service_name* is the name to give the service. If you do not specify a *service_name*, the service uses the default name “ims54alertsvc.” You can have multiple installations.
- -setup *service_name* – change the parameters of the service.

- -home *dir* – installs the service in the location specified by the *dir* parameter, which should be the fully qualified path to the working directory.
- -file *cfg* – creates the alert configuration file using the name specified by the *cfg* parameter.
- -start *service_name* – start the service.

Note The service is installed with a "Manual" startup type, which means you must use the -start command to launch the service. To have the service start automatically each time you start or reboot your machine, select Start | Settings | Control Panel | Administrative Tools | Services, right-click the alerts service in the Services window right pane, then select Properties. When the Properties window opens, select Automatic from the Startup Type drop-down list. Click OK to save the change and close the window.

- -stop *service_name* – stop the service.
- -restart *service_name* – restart the service.
- -remove *service_name* – remove the service. Use ims alertdsvc -install service_name to reinstall it.

Note You can also use the Windows Services applet (Start | Settings | Control Panel | Administrative Tools | Services) to start, stop, pause, resume, and restart the service. When the Services window opens, right-click the alert service and select the command you want to execute.

Troubleshooting the alerts service

The following notes may be helpful when troubleshooting system problems:

- The alerts service writes to a log file named *_ims54alertd.log* in *x:\Sybase\ImpactServer-5_4\bin\.bin* at start up and shut down. If you have problems with the alerts service, examine the log file first. A common error is an invalid home directory specified during the service's install or set up.
- If the alert service fails to start and the log file is not created, the most common reason is that the required environment variables are not properly defined, which prevents alertdsrv from executing. This error may be caused because the script or binary cannot find shared libraries.

- After a successful startup, alertdsrv creates *alertd.xlog1* in the home directory specified during install or setup. The following shows typical log entries for a successful startup:

```
2005-06-14 15:18:03.807 Log limit            = 1000000
2005-06-14 15:18:03.807 Log max roll depth   = 2
2005-06-14 15:18:03.807 | ims54alertd | 2524 | 1124 |  |

BEGIN_COPYRIGHT
Confidential property of Sybase, Inc.
Copyright 1987-2005.
Sybase, Inc. All rights reserved.

Unpublished rights reserved under U.S. copyright laws.
END_COPYRIGHT

BEGIN_DISCLAIMER
This software contains confidential and trade secret information of
Sybase, Inc. Use, duplication or disclosure of the software and
documentation by the U.S. Government is subject to restrictions set
forth in a license agreement between the Government and Sybase, Inc.
or other written agreement specifying the Government's rights to use
the software and any applicable FAR provisions, for example,
FAR 52.227-19.

Sybase, Inc. One Sybase Drive, Dublin, CA 94568, USA
END_DISCLAIMER

2005-06-14 15:18:03.807 | ims54alertd | 2524 | 1124 |  | Alert handler
     start
2005-06-14 15:18:03.807 | ims54alertd | 2524 | 1124 |  | Create context:
     OTI_Context
2005-06-14 15:18:03.807 | ims54alertd | 2524 | 1124 |  | Create input
     transport: AlertsIn
2005-06-14 15:18:03.817 | ims54alertd | 2524 | 1124 |  | Create log
     transport: AlertLog
2005-06-14 15:18:03.817 | ims54alertd | 2524 | 1124 |  | Create error
     transport: AlertError
2005-06-14 15:18:03.817 | ims54alertd | 2524 | 1124 |  | Will poll
     transport at rate: 3 s
2005-06-14 15:18:03.817 | ims54alertd | 2524 | 1124 |  | Enter read/poll
     loop...
2005-06-14 15:18:03.827 | ims54alertd | 2524 | 1124 |  | Started Command
     "myalert", in dir: ".", PID: 2460
2005-06-14 15:18:10.818 | ims54alertd | 2524 | 1124 |  | Finished Command
     "myalert", PID 2460, exit code: 0
2005-06-14 15:18:10.818 | ims54alertd | 2524 | 1124 |  | Move Alert
```

```
     message to log transport
```

A typical sequence for a successful read, the invocation of a script or binary, and log creation looks similar to this:

```
2005-06-14 15:18:03.827 | ims54alertd | 2524 | 1124 |  | Started Command
     "myalert", in dir: ".", PID: 2460
2005-06-14 15:18:10.818 | ims54alertd | 2524 | 1124 |  | Finished Command
     "myalert", PID 2460, exit code: 0
2005-06-14 15:18:10.818 | ims54alertd | 2524 | 1124 |  | Move Alert
     message to log transport
```

The following code shows typical log entries for a successful shutdown when alerts are running as a Windows service:

```
2005-06-14 15:18:10.818 | ims54alertdsvc | 2524 | 1124 |  | Alert handler
     exiting
```

- If the script or binary blocks awaiting user input, such as for a pause or read command, the “Finished…” and “Move…” log lines are not present. To break the deadlock, use the Windows Task Manager to end the script or binary process. You should then see these log entries.

Running alerts as a UNIX daemon

To run alertd as a daemon, type `ims alertd` in a terminal window, which returns a shell prompt and does not exit when your login session terminates.

To run alertd as a daemon at startup, consult your UNIX documentation.

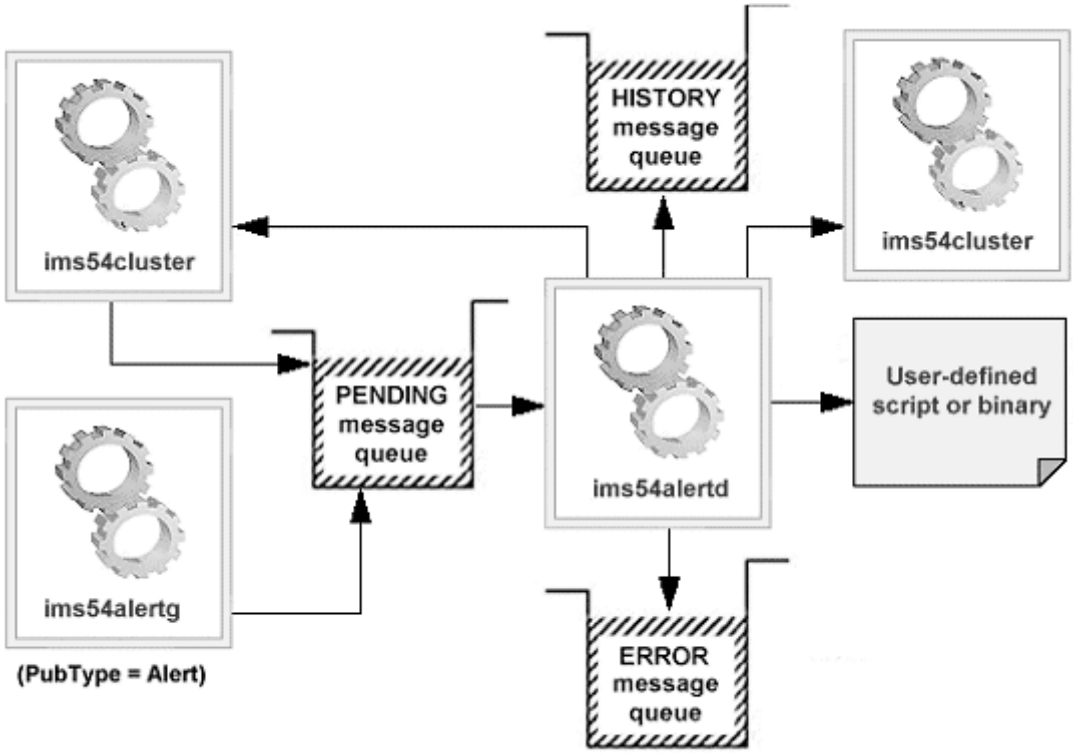
Using alerts

To use the alert information from ims54cluster or alertg, an OT-XML message on the alert transport, such as an item in the PENDING queue, must be turned into a user-defined action. Use the utility program, alertd, to do this.

alertd reads alert messages from the alert transport, such as a queue or file device, and invokes a user-maintained shell script or binary. Use alertdsvc to run as a Windows service. On UNIX systems, run alertd either in the foreground or in the background, for example, as a daemon.

Before invoking the user script or binary, alertd places the relevant information from the alert message in environment variables. While the user script or binary is executing, you can query the environment variables to control the execution flow in a script or binary.

Figure 5-2: Open Transport-XML alerts



Alerts automation

To automate the treatment of XML-alerts,

- XML alerts must be dropped into an OT queue using ims54cluster or the alertg tester application.
- alertd must be configured to listen for alerts.
- A user-maintained script must be configured for alertd (*.ALERTS.COMMAND*, *.ALERT.ISSCRIPT*, and *.ALERT.EXECDIR*).

For each XML alert picked up from the OT queue by alertd:

- alertd sets all XML alerts elements to predefined environment variables.
- alertd invokes the configured user-maintained shell script or binary in a separate process.

The user-maintained script is responsible for automating and handling the response to the alert. It can for example, page an operator depending on the alert severity, send an e-mail, log the alert into a database for warehousing, execute file or log maintenance tasks, and so on.

e-Biz Impact provides the sample *myalert.sh* script that demonstrates how to read the environment variables and dump the elements of the alert to the console output. *myalert.sh* is located in *x:\Sybase\ImpactServer-5_4\samples\alerts* on Windows and in *~/Sybase/ImpactServer-5_4/samples/alerts* on UNIX, where "*x*" and "~" represent the location where the e-Biz Impact server is installed.

```
#!/usr/bin/sh
# Impact Alert Handler Script
# Utilize the alert* environment variables as
# appropriate for your site.
echo alertVersion = $alertVersion
echo alertServerName = $alertServerName
echo alertDomainName = $alertDomainName
echo alertDomainType = $alertDomainType
echo alertClusterName = $alertClusterName
echo alertClusterConfigID = $alertClusterConfigID
echo alertEventTime = $alertEventTime
echo alertPublisherName = $alertPublisherName
echo alertPublisherType = $alertPublisherType
echo alertPublisherSessionID = $alertPublisherSessionID
echo alertPublisherInstance = $alertPublisherInstance
echo alertObjectName = $alertObjectName
echo alertObjectType = $alertObjectType
echo alertObjectSessionID = $alertObjectSessionID
echo alertObjectInstance = $alertObjectInstance
echo alertReasonCode1 = $alertReasonCode1
echo alertReasonCode2 = $alertReasonCode2
echo alertReasonCode3 = $alertReasonCode3
echo alertReasonCode4 = $alertReasonCode4
echo alertReasonText = $alertReasonText
echo alertTextEncoding = $alertTextEncoding
echo
---------------------------------------
# In production, handler scripts should not block on
```

```
# user input since it prevents the alert handler daemon
# from processing other alerts.
# Pausing can be useful during development. On Windows,
# make sure you set Alerts.Visible to TRUE so you can
# interact (e.g. respond to pause).
echo press Enter/Return to continue
read xxx
```

Sample configuration files and alert-handler scripts

A complete set of alert configuration files as well as sample shell scripts are distributed with e-Biz Impact. The sample configuration files are located in:

```
~/Sybase/ImpactServer-5_4/samples/alerts
```

This section describes how to use the distributed sample configuration files and alert handler scripts. If you are using the sample Web OT configuration file, make sure the QMALERTS queue manager has been created with the queues PENDING, HISTORY, and ERROR. Also make sure that the WebSphere MQ queue manager is running.

❖ **Running the demonstration alert application**

1 Go to the *~/Sybase/ImpactServer-5_4/samples/alerts* on UNIX and *x:\Sybase\ImpactServer-5_4\samples\alerts* on Windows.

2 Type `ims alertg` at a command line in Windows or in a terminal window on UNIX. You should see:

```
Alert/Trap publication test application
Constructing Alert...
Starting Alert support...
Sending Alert...
Sent!
Stopping Alert support...
Done.
```

3 Type ims alertd at a command line in Windows. In UNIX, enter `ims alertd -shell` in the terminal window. You should see a command window display with this content:

```
alertVersion = 1
alertServerName = myhost
alertDomainName = myImpact
alertDomainType = myTest
alertClusterName = myCluster
alertClusterConfigID = 999
```

```
alertEventTime = 2003-01-09T19:00:48
alertPublisherName = foobar
alertPublisherType = sfm
alertPublisherSessionID =
alertPublisherInstance = 2
alertObjectName = foobar
alertObjectType = sfm
alertObjectSessionID =
alertObjectInstance = 4
alertReasonCode1 = Refuse mode
alertReasonCode2 = Console
alertReasonCode3 = Fred
alertReasonCode4 =
alertReasonText = SFM 'foobar' was put in Refuse
mode by the Console by Fred
alertTextEncoding = ANSI
---------------------------------------
Press any key to continue . . .
```

4 Press Enter to continue and close the window.

5 Press Ctrl-C to exit *alertd*.

This is what happened:

1 alertg placed the alert data into a message and placed it on the Transport.AlertsIn device (PENDING queue or *.ALERTS_PENDING* file).

2 alertd read that message from the same device, and invoked the sample script, for example, *myscript.cmd* for Windows or *myscript.sh* for UNIX.

3 The script dumped out the environment variables set by alertd, and then paused, waiting for user input.

4 After pressing Enter, alertd goes back to waiting for more alert messages.

5 When you pressed Ctrl-C, alertd exited.

In one session, you can perform variations of the preceding steps by running alertd. In another session, run alertg multiple times while alertd is running.

Scripts or binaries should not generally block awaiting user input, but for the purposes of the demonstration, *myscript* paused for you to see the result; for example, all the echo statements that displayed the environment variables set by alertd.

In general, debug your handler scripts or binaries using foreground mode. When all of your scripts are debugged, run alertd as a Windows service or as a UNIX daemon (see "Configuring alertd to run as a service or daemon" on page 95). In the alertd configuration file, set Alerts.Visible to FALSE before running alertd as a service or daemon.

Sending OT-XML alerts to a cluster

In addition to sending OT-XML alerts to alertd from WebSphere MQ messaging queues, e-Biz Impact 5.4 also allows a cluster to access OT-XML alerts from messaging queues using ODL.

To enable e-Biz Impact ODL to access WebSphere MQ message queues you:

1 Configure *nnsyreg.dat* to define transport properties.

2 Develop and configure an ODL acquisition AIM (MQAcq) to acquire data from WebSphere MQ message queues. Use clot objects to define OT instances and implement OT calls through ODL, then define these instances to access the desired data.

3 Develop and configure an ODL router application.

MQAcq and MQDel applications work with router applications, rather than an SFM, to route messages or alerts to or from WebSphere MQ message queues. A router is an SFM running in "router only" mode. Configuration options for router properties are similar or identical to SFM properties. However, no log file maintenance is provided for routers; the router is responsible only for synchronous routing of the message, not storage.

4 Develop and configure an ODL delivery AIM (MQDel) that accepts a transaction from a router and delivers it to the specified WebSphere MQ message queue. You must also copy the *<install directory>/ImpactServer-5_4/include/IMPACT/MQBRIDGE/mqdel.prj* to the working directory for the application instance. A sample *nnsyreg.dat* file is also provided in the directory.

Additional information

For more information, see the following books on the e-Biz Impact SyBooks CD:

- Open Transport documentation

- *e-Biz Impact Application Guide*, Chapter 4, “Accessing MQSeries Data”
- *e-Biz Impact Configuration Guide*, Chapter 4, “Configuring Applications”

APPENDIX A

Alert IDs

This appendix identifies the types of alerts that can be generated from e-Biz Impact and lists their associated identification numbers.

Controller alerts

Table A-1: Controller alert IDs

ID number	Alert name
20101	IMPACT_ALERT_ID_CONTROLLER_START_FAILURE
20102	IMPACT_ALERT_ID_CONTROLLER_START_SUCCESS
20110	IMPACT_ALERT_ID_CONTROLLER_RESTART
20120	IMPACT_ALERT_ID_CONTROLLER_DEATH
20130	IMPACT_ALERT_ID_CONTROLLER_DISABLE
20141	IMPACT_ALERT_ID_CONTROLLER_KILL_REQUEST
20143	IMPACT_ALERT_ID_CONTROLLER_KILL_SUCCESS
20151	IMPACT_ALERT_ID_CONTROLLER_SHUTDOWN_REQUEST
20152	IMPACT_ALERT_ID_CONTROLLER_SHUTDOWN_TIMEOUT
20153	IMPACT_ALERT_ID_CONTROLLER_SHUTDOWN_SUCCESS
20161	IMPACT_ALERT_ID_CONTROLLER_RELOAD_REQUEST

Application alerts

Table A-2: Application alert IDs

ID number	Alert name
30101	IMPACT_ALERT_ID_APPLICATION_INSTANCE_START
30102	IMPACT_ALERT_ID_APPLICATION_INSTANCE_STOP

SFM alerts

Note All the alerts in the Router mode have values of 700XX instead of 600xx.

Table A-3: SFM alert IDs

ID number	Alert name
60001	Error opening fkey history file (NT)
60002	Error opening fkey history file mapping (NT)
60003	Error opening fkey history file mapping view (NT)
60004	Unknown error opening existing fkey history file (Unix)
60005	Error opening existing fkey history file (Unix)
60006	Error stat'ing fkey history file (Unix)
60007	Error mapping fkey history file (Unix)
60008	Error unmapping fkey history file (NT)
60009	Error closing memory map handle (NT)
60010	Error closing fkey history file handle (NT)
60011	Error unmapping fkey history file (Unix)
60012	Error closing fkey history file (Unix)
60013	Error registering SNMP source info
60014	Error registering SNMP route info
60015	Error registering SNMP prod info
60016	Error registering SNMP daemon info
60017	Error registering SNMP dest info (Router mode)
60018	Error registering dynamic SNMP source info
60019	Dynamic SNMP source info created
60020	Ping source failed
60021	Error opening completed/unroutable log file
60022	Error opening archive

ID number	Alert name
60023	Error in archiving during startup
60024	Error opening pending log file (NT)
60025	Error opening pending log file (Unix)
60026	Error creating handles for pending log file (NT)
60027	Error creating handles for pending log file (Unix)
60028	Bogus destination created
60029	Pending log file name not defined
60030	Pending log file path error
60031	Pending log file size not defined
60032	Pending log file size invalid
60033	Unroutable log file name not defined
60034	Unroutable log file path error
60035	Completed log file path error
60036	Completed log file size not defined
60037	Completed log file size invalid
60038	Archive name not defined
60039	Archive size not defined
60040	Archive size invalid
60041	Dest reaches maximum retry count
60042	Dest starts retry
60043	Qual ODL object Init failed
60044	Qual ODL object Init Rte() failed
60045	Tran ODL object Init failed
60046	Tran ODL object Init Rte () failed
60047	Error registering SNMP SFM info
60048	Qual ODL object Deinit failed
60049	Tran ODL object Deinit failed
60050	Router mode cannot service regular DFC (x2)
60051	SFM cannot service route_sync (not in Router mode)
60052	Received unknown DFC
60053	Error extracting DFC arguments
60054	Entry audit prod error
60055	Error qualifying transaction
60056	Tran recycle bad return value
60057	Tran recycle failed
60058	Submit/SubmitTran bad return value
60059	Submit/SubmitTran failed

ID number	Alert name
60060	Transaction stuck (filter/prod obj property)
60061	Transaction cancelled (filter/prod obj property)
60062	Transaction skipped (filter/prod obj property)
60063	Exit audit prod error
60064	DfcSend() Error (x3)
60065	Dfc bus error (DfcSend() errno) (x3)
60066	AYT failed (bad return value)
60067	PING failed (bad return value)
60068	Pending log file high water mark
60069	Completed log file high water mark
60070	Pending log file full
60071	SFM in refuse mode when receiving transaction
60072	SFM unknown error (x2)
60073	Unroutable transaction (Bad route)
60074	Unroutable transaction (Recycled prod exclusion)
60075	Unroutable transaction (Does not quai)
60076	Router mode dispatch failed
60077	Fkey duplicate
60078	Fkey non-commit
60079	Fkey invalid source
60080	Fkey too long
60081	CNC command not supported in Router mode
60082	Unknown CNC command
60083	SFM enters refuse mode
60084	SFM enters accept mode
60085	SigEvent logfile path error
60086	LastID file path error
60087	LastID file open error
60088	LastID file write error
60089	Fkey file path error

Index

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W