

Progress Fathom Management Standard Edition

Database Management Guide

Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. This manual is also copyrighted and all rights are reserved. This manual may not, in whole or in part, be copied, photocopied, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Progress Software Corporation.

The information in this manual is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear in this document. The references in this manual to specific platforms supported are subject to change.

Allegrix, A [Stylized], ObjectStore, Progress, Powered by Progress, Progress Fast Track, Progress Profiles, Partners in Progress, Partners en Progress, Progress en Partners, Progress in Progress, P.I.P., Progress Results, ProVision, ProCare, ProtoSpeed, SmartBeans, SpeedScript, and WebSpeed are registered trademarks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and/or other countries. AccelEvent, A Data Center of Your Very Own, Allegrix & Design, AppsAlive, AppServer, ASPen, ASP-in-a-Box, BusinessEdge, Business Empowerment, Empowerment Center, eXcelon, Fathom, Future Proof, IntelliStream, ObjectCache, OpenEdge, PeerDirect, POSSE, POSSENET, Progress Business Empowerment, Progress Dynamics, Progress Empowerment Center, Progress Empowerment Program, Progress for Partners, Progress OpenEdge, Progress Software Developers Network, PSE Pro, PS Select, SectorAlliance, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Technical Empowerment, WebClient, and Who Makes Progress are trademarks or service marks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and other countries.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Any other trademarks or service marks contained herein are the property of their respective owners.

Fathom Management includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Copyright © 1999 The Apache Software Foundation. All rights reserved (Xalan XSLT Processor) and Copyright © 2000-2002 The Apache Software Foundation. All rights reserved (Jakarta-Oro). The names “Apache,” “Xerces,” “Jakarta-Oro,” and “Apache Software Foundation” must not be used to endorse or promote products derived from this software without prior written permission. Products derived from this software may not be called “Apache” or “Jakarta-Oro,” nor may “Apache” or “Jakarta-Oro” appear in their name, without prior written permission of the Apache Software Foundation. For written permission, please contact apache@apache.org. Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

Fathom Management includes software developed by ACME Labs. Copyright © 2000 by Jef Poskanzer <jef@acme.com>. All rights reserved. Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

Fathom Management includes software developed by Sun Microsystems, Inc. Copyright © 2003 Sun Microsystems, Inc. All Rights Reserved. Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

Fathom Management includes the Jetty Package Copyright © 1998 Mort Bay Consulting Pty. Ltd. (Australia).

Fathom Management includes software developed by the ModelObjectsGroup (<http://www.modelobjects.com>). Copyright © 2000-2001 ModelObjects Group. All rights reserved. The name “ModelObjects” must not be used to endorse or promote products derived from the SSC Software without prior written permission. Products derived from the SSC Software may not be called “ModelObjects”, nor may “ModelObjects” appear in their name, without prior written permission. For written permission, please contact djacobs@modelobjects.com.

Fathom Management includes files that are subject to the Netscape Public License Version 1.1 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at (<http://www.mozilla.org/NPL>). Software distributed under the License is distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License. The Original Code is Mozilla Communicator client code, released March 31, 1998. The Initial Developer of the Original Code is Netscape Communications Corporation. Portions created by Netscape are Copyright © 1998-1999 Netscape Communications Corporation. All Rights Reserved.

Fathom Management contains copyright material licensed fromAdventNet, Inc. <http://www.adventnet.com>. All rights to such copyright material rest with AdventNet.

Fathom Management includes the RSA Data Security, Inc. MD5 Message-Digest Algorithm. Copyright © 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

August 2004



Product Code: 4496

Release: V3.0A

Contents

Preface	Preface–1
1. Database Resource Monitoring.	1–1
Resource monitoring terms	1–2
Database management concepts and terminology	1–3
Managed versus scripted databases.	1–3
Database migration	1–3
Starting a remote managed database agent	1–4
Monitoring managed databases	1–5
Default database monitoring plans	1–5
Modifying database monitoring plans	1–7
Monitoring scripted databases	1–9
Starting a scripted database agent	1–10
Understanding and using database rules	1–11
Working with database rule sets	1–14
Creating a database rule set	1–14
Editing a database rule set	1–15
Copying a database rule set	1–15
Deleting a database rule set	1–16
Rule sets with one or more rules in common	1–16
Associating a rule set with a database monitoring plan.	1–16
Using the Configuration Advisor	1–17
Understanding the recommended threshold settings	1–21
Comparing and selecting threshold settings	1–22
Submitting your threshold setting selections	1–23
Determining the effectiveness of your selections	1–23

2.	Graphical Displays of Database Data	2-1
	Interpreting graphical database data	2-2
	Types of graphs	2-3
	General graph information	2-4
	Display characteristics of a pinup graph	2-6
3.	Examining Data from an OpenEdge Database	3-1
	Accessing database information	3-2
	Changing database control settings	3-3
	Database log file monitors	3-6
	Log file monitor default values	3-6
	Log file monitor considerations	3-7
	Customizing a database log file monitor	3-7
	Using the log file viewer	3-8
	Understanding OpenEdge database views	3-11
	Vital Signs view	3-13
	File Systems view	3-15
	Storage Area views	3-18
	User Activity views	3-26
	Record and Index Activity view	3-28
	Locks and Latches views	3-31
	Transactions view	3-32
	Memory Resources view	3-33
	Page Writers view	3-35
	Start Parameters view	3-37
	General Details view	3-38
	Advanced views	3-38
4.	Database Maintenance Job Templates	4-1
	Database maintenance job templates	4-2
	Displaying database job templates	4-2
	General considerations	4-3
	Creating database maintenance job instances	4-4
	Database Backup jobs	4-5
	Database Restore job	4-8
	Truncate BI job	4-10
	Grow BI job	4-12
	Database Analysis job	4-14
	Index Compaction job	4-15
	Backup Configuration job	4-16
	Advanced information	4-18
	Environment variables	4-19

Index	Index-1
--------------	----------------

Figures

Figure 1–1:	Database monitoring plan default values	1–5
Figure 1–2:	Available Rules page	1–12
Figure 2–1:	Graph legend	2–2
Figure 2–2:	Full meter graph	2–3
Figure 2–3:	Hi/low graph	2–4
Figure 2–4:	Graph with threshold indicated	2–4
Figure 3–1:	Database Details page	3–2
Figure 3–2:	Managed Database Control page	3–4
Figure 3–3:	Scripted Database Control page	3–4
Figure 3–4:	Vital Signs view page (1 of 2)	3–13
Figure 3–5:	Vital Signs view page (2 of 2)	3–14
Figure 3–6:	File Systems view (1 of 2)	3–15
Figure 3–7:	File Systems view (2 of 2)	3–16
Figure 3–8:	Storage Area view (1 of 3)	3–18
Figure 3–9:	Storage Area view (2 of 3)	3–19
Figure 3–10:	Storage Area view (3 of 3)	3–19
Figure 3–11:	Control Area view	3–21
Figure 3–12:	Order Area summary view	3–22
Figure 3–13:	Additional Order Area view	3–23
Figure 3–14:	Before Image Primary Recovery Area view	3–24
Figure 3–15:	After Image Storage Area view	3–25
Figure 3–16:	User Activity view	3–26
Figure 3–17:	User details	3–27
Figure 3–18:	Transactions and locks	3–28
Figure 3–19:	Record and Index Activity view	3–29
Figure 3–20:	Record/Table information	3–29
Figure 3–21:	Index information	3–30
Figure 3–22:	Lock summary section	3–31
Figure 3–23:	Latch summary section	3–31
Figure 3–24:	Transactions page	3–32
Figure 3–25:	Buffers summary section	3–33
Figure 3–26:	Shared memory summary section	3–34
Figure 3–27:	Space management summary section	3–34
Figure 3–28:	Resources summary section	3–34
Figure 3–29:	Before Image summary section	3–35
Figure 3–30:	After Image summary section	3–35
Figure 3–31:	Asynchronous Page Writer summary section	3–35
Figure 3–32:	Checkpoint summary section	3–36
Figure 3–33:	Database Start Parameters page	3–37
Figure 3–34:	General Details view	3–38
Figure 3–35:	Raw VST Data page	3–38

Figure 3–36: _ActAllLog Data page 3–39

Figure 3–37: Raw System Table Data page 3–39

Figure 3–38: Sample Raw Table Data page 3–40

Figure 4–1: Database maintenance job templates 4–2

Tables

Table 1–1:	Monitoring plan default values	1–6
Table 1–2:	Resource status legend	1–13
Table 2–1:	Graph properties and options	2–6
Table 3–1:	Documentation about log files	3–9
Table 3–2:	Database views	3–12
Table 4–1:	Environmental variables for database maintenance jobs	4–19

Preface

This Preface contains the following sections:

- [Purpose](#)
- [Audience](#)
- [Fathom Management with OpenEdge or Progress](#)
- [Organization](#)
- [Typographical conventions](#)

Purpose

This guide describes how to use Progress® Fathom™ Management Standard Edition to manage and monitor the condition of your OpenEdge™ databases. Concepts explained in this guide include: scripted and managed databases, database monitoring plans, database views, and database maintenance jobs.

Throughout this guide, Fathom Management Standard Edition is referred to as Fathom.

Audience

This guide is designed for users of the Fathom Management product. Typical users are OpenEdge database administrators and any others responsible for the daily management of a database. Before reading this guide, you should be familiar with the Fathom Management UI and with the basic concepts of resource monitoring. See the [Resource Monitoring Guide](#) for an explanation of the Fathom UI and for an introduction to resource monitoring.

Fathom Management with OpenEdge or Progress

Fathom Management Version 3.0A runs against the following:

- OpenEdge 10.0B.
- Progress Version 9.1D and the 9.1D09 service pack.

For the sake of simplicity, the procedures and screen shots provided in this manual refer to running Fathom against OpenEdge 10.0B. Unless indicated otherwise, the procedures are the same for both Progress Version 9.1D with the 9.1D09 service pack and OpenEdge 10.0B. For example, if a procedure refers to an OpenEdge database, the procedure applies to a Progress database as well.

Organization

Chapter 1, “Database Resource Monitoring”

Provides details about how Fathom recognizes a database’s existence, how to set up default database monitoring, how to use rules with a database, and how to update database monitoring plans. Use the information in this chapter in conjunction with the information provided in the *Resource Monitoring Guide*.

Chapter 2, “Graphical Displays of Database Data”

Describes the graphical display of database data, including pinup graphs.

Chapter 3, “Examining Data from an OpenEdge Database”

Presents information about database log file monitors, database log file monitoring plans and rules, and database views.



Chapter 4, “Database Maintenance Job Templates”

Describes the Fathom-provided job templates you use to set up and schedule routine maintenance activities for your Progress database.

Typographical conventions

This manual uses the following typographical conventions:

Convention	Description
Bold	Bold typeface indicates commands or characters the user types, or the names of user interface elements.
<i>Italic</i>	Italic typeface indicates the title of a document, provides emphasis, or signifies new terms.
SMALL, BOLD CAPITAL LETTERS	Small, bold capital letters indicate OpenEdge™ key functions and generic keyboard keys; for example, GET and CTRL .
KEY1-KEY2	A hyphen between key names indicates a <i>simultaneous</i> key sequence: you press and hold down the first key while pressing the second key. For example, CTRL-X .

Convention	Description
KEY1 KEY2	A space between key names indicates a <i>sequential</i> key sequence: you press and release the first key, then press another key. For example, ESCAPE H .
Syntax:	
Fixed width	A fixed-width font is used in syntax statements, code examples, and for system output and filenames.
<i>Fixed-width italics</i>	Fixed-width italics indicate variables in syntax statements.
Fixed-width bold	Fixed-width bold indicates variables with special emphasis.
UPPERCASE fixed width	Uppercase words are Progress® 4GL language keywords. Although these always are shown in uppercase, you can type them in either uppercase or lowercase in a procedure.
	This icon (three arrows) introduces a multi-step procedure.
	This icon (one arrow) introduces a single-step procedure.

Database Resource Monitoring

By monitoring the databases in your environment and alerting you to issues of potential concern, Progress® Fathom™ Management helps you better manage your database resources. This chapter introduces resource monitoring terms and explains how Fathom uses rules and schedules to manage database resources, as described in the following sections:

- [Resource monitoring terms](#)
- [Database management concepts and terminology](#)
- [Database migration](#)
- [Monitoring managed databases](#)
- [Monitoring scripted databases](#)
- [Understanding and using database rules](#)
- [Working with database rule sets](#)
- [Using the Configuration Advisor](#)

Note: The procedures and screen shots provided in this manual are of Fathom running against OpenEdge™ 10.0B.

Resource monitoring terms

Before you begin monitoring your databases, it is important to understand the different pieces of resource monitoring and how they work together. A *resource* is a specific component of your configuration.

Fathom monitors the following resource types:

- Database
- File
- Network
- Progress® OpenEdge™ Server products
- System

Terms used by Fathom for resource monitoring include:

- **Actions** — Activities triggered in response to alerts. For example, you might specify that you receive an e-mail if a database crashes.
- **Alerts** — Notifications sent when rules are broken.
- **Resource monitoring plan** — A plan that identifies rules against which the resource is checked and the schedule during which the resource is checked. Fathom cannot monitor a resource without a monitoring plan.
- **Rules** — Criteria against which a resource's performance is measured.
- **Schedule** — The block of time during which a resource is polled.

This guide focuses on database resource monitoring. See the [Resource Monitoring Guide](#) for information about monitoring file, network, and system resources. See the [OpenEdge Server Management Guide](#) for more information about monitoring OpenEdge servers.

Database management concepts and terminology

Fathom Management runs as a process or thread in the AdminServer. Therefore, the type of relationship that you can establish between a database and Fathom Management depends on the relationship that currently exists between a database and the AdminServer that Fathom is running in.

Managed versus scripted databases

A database recognized and managed by the AdminServer is called *managed*.

A database not managed by an AdminServer is called *scripted*. Scripted databases are administered outside the AdminServer using parameter files (.pf) and operating system-dependent scripts.

You can elect to migrate a scripted database to a managed database. To migrate a scripted database to a managed database, perform the steps in the “[Database migration](#)” section on page 1–3.

Database migration

After you initially configure Fathom, you can use the Database Migration utility at any time to migrate a scripted database to a managed database.



To migrate a database from the Fathom Management console:

1. Click **Resources**. The **Resources** page appears.
2. Click **Database Migration Utility**. The **Database Information** page appears. This page contains two sections. The first section provides an overview of what information you must provide to migrate the database, and the second allows you to provide specifics about the particular database you want to migrate.
3. Enter the database’s information into the second section of the **Database Information** page and click **Submit**.

Note: For additional details about the information you must provide when migrating a database, see the [Installation and Configuration Guide](#).

The AdminServer now controls the database. Use the AdminServer or the DBMAN utility to start and stop the database. If you managed the database with scripts before its migration, do not use these scripts. If you use these scripts, Fathom will not recognize your database. Fathom can only recognize databases run by the AdminServer.

Starting a remote managed database agent

If the AdminServer managing the remote database is configured as a container, you do not need to take any steps to start the database agent. It will autostart whenever the broker starts.

However, if the AdminServer is not configured as a container, you must edit the `conmgr.properties` file to start the database agent for a remote, managed database.



To enable a remote managed database agent:

1. Shut down the AdminServer.
2. Open the database's `conmgr.properties` file. This file is located in *OpenEdge-install-directory/properties/conmgr.properties*.
3. Add the following to the `conmgr.properties` file:

```
[agentremoteconnection]
agentremotesupport=true
```

4. Save the `conmgr.properties` file.
5. Restart the AdminServer.
6. Open Progress Explorer.
7. Select the default configuration for the database.
8. On the agent folder, specify the Fathom host name and the database admin port.

Note: Remote database agents do not reconnect to Fathom if the machine hosting Fathom is rebooted after the AdminServer is shutdown. After rebooting the machine hosting Fathom, you must restart remote database agents.

Monitoring managed databases

Before it can monitor a database, Fathom needs a monitoring plan. Fathom automatically creates a default monitoring plan for each managed database. The default monitoring plan contains a default rule set and schedule. You can change the default values at any time, or create a new monitoring plan.

Default database monitoring plans

Fathom provides a default monitoring plan for each managed database. [Figure 1–1](#) shows a sample default database monitoring plan.

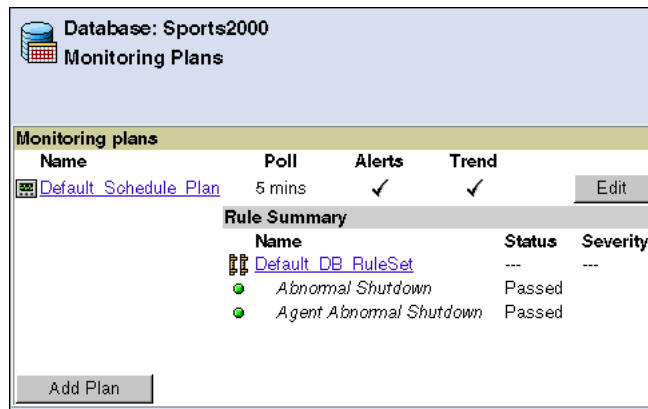


Figure 1–1: Database monitoring plan default values

The default monitoring plan shown in [Figure 1–1](#) consists of the values described in [Table 1–1](#).

Table 1–1: Monitoring plan default values

Field	Default value	Description
Schedule	Default schedule plan	Identifies the system-defined 24/7 default schedule used when the plan is active.
Poll	5 mins	Identifies the polling cycle, which is the frequency at which the resource’s rules are checked, set up for this database resource monitor.
Alerts	Enabled	Indicates whether alerts are active and will be generated when the plan is active.
Trend	Disabled	Indicates whether the data monitored while this plan is active will (or will not) be stored to the FathomTrendDatabase. (This option will not be initially set as a default value unless you chose it when you added the database to Fathom.)
Rule summary	Default DB rule set	Consists of two rules: Abnormal Shutdown and AgentAbnormalShutdown.

Note: Fathom prevents the assignment of schedules that share overlapping time periods. For example, if you have a Default Schedule set up for a resource monitor, you cannot set up an additional plan because the Default_Schedule is defined for 7 days a week, 24 hours a day. You must modify the Default_Schedule or remove it from the plan in order to add any other plans.

Modifying database monitoring plans

You can modify the default database monitoring plan or any database monitoring plans you create.



To modify a database monitoring plan:

1. Click **Resources** in the Fathom Management console menu bar. Expand the entries in the list frame to list your database resources.
2. Click the name of the specific database whose monitoring plan you want to update. The database page for the selected database appears in the detail frame:

The screenshot shows the Fathom Management console interface for a specific database resource. At the top, the title bar reads "Database:DEVPC01.Sports2000" with a "Delete" button. Below this, the "Database status" section shows "Database: Running" and "Monitoring Agent: Running". The "Command and control" section contains links for "Control" (Start or stop database), "Maintenance Jobs" (Maintenance jobs associated with this database), "Log File Monitor" (Log file errors on which to alert), "Monitoring Plans" (Database conditions on which to alert), "Configuration Advisor" (Suggests rule thresholds based on data in the trend database), and "Log File Viewer" (Examine the DB's log file). The "Operational views" section includes "Vital Signs" (Summary of Important DB Information), "Storage Areas" (Storage Area and extent information and statistics), "Record and Index Activity" (I/O activity for tables and indexes), "Transactions" (Database transactions), "Page Writers" (Summary of Page Writer Activities), "File Systems" (Monitor Database File Systems), "User Activity" (Connected users & user statistics), "Locks and Latches" (Lock & Latch Activity Information), and "Memory Resources" (Summary of Buffers, Memory, etc.). The "Informational views" section includes "Start Parameters" (Start parameters of the DB) and "General Details" (Start info, Backup info, other general DB info). The "Advanced views (raw data)" section includes "Raw VST Data" (Examine individual VST tables) and "Raw System Table Data" (Examine individual System tables).

Note that the name of the database resource is displayed, preceded by the container name. In this example, the **Sports2000** database is contained by **DEVPC01**.

3. Click **Monitoring Plans** in the **Command and control** section of the **Database Details** page. The monitoring plans for the specified database appear.
4. Click **Edit** for the schedule associated with the plan that you want to update. The figure below shows the page that appears when the **Default Schedule Plan** is selected:

Edit Default_Schedule Monitoring Plan for: Sports2000

Save Cancel

Monitoring plan definition

Available Schedules: Default_Schedule

Polling Interval: 5 minutes

Alerts Enabled: ☒

Trend Performance Data: ☒ Advanced Settings

Rules selected for this plan

Name	Status	Severity
Default_DB_RuleSet	---	---
Abnormal Shutdown	Passed	
Agent Abnormal Shutdown	Passed	

Add Rule Select Rule Sets

5. Edit the values in the **Monitoring plan definition** section as needed. To see current trend value settings, click **Advanced Settings**.
6. Click **Add Rule** in the **Rules selected for this plan** section of the page. The **Available Rules** page appears. See the “[Understanding and using database rules](#)” section on page 1–11 for more information on database rules.
7. Click the rule you want to add and update the values you want to edit. Click **Save**. The **Available Rules** page reappears.

Note: Any rules you define and add are associated only with this particular plan. If you create another plan and add the same rules, you can select values that are appropriate for that particular plan.

8. Repeat [Step 7](#) for each additional rule you want to apply to this plan. After you add and define the criteria for each rule you want to add, click **Done Adding Rules** on the **Available Rules** page. The **Edit** page for your schedule reappears.

To add any rule sets to your plan, click **Select Rule Sets** in the **Rules selected for this plan** section of the page. See the “[Associating a rule set with a database monitoring plan](#)” section on page 1–16 for information about adding rule sets.

9. Click **Save**. The updated monitoring plan appears in the monitoring plan definition on the top of the **Monitoring Plan summary** page.

Monitoring scripted databases

To create a monitoring plan for a scripted database, you must perform two tasks:

- Add a resource monitor for a scripted database.
- Start a dbagent for the scripted database. For details about how to enable a scripted database agent, see the “[Starting a scripted database agent](#)” section on page 1–10.



To create a resource monitoring plan for a scripted database:

1. From the Fathom Management console, click **Resources**.
2. Click **New Resource Monitor**. The **New Resource Monitor** page appears.
3. Click **Scripted Database**. The **Create Scripted Database Monitor** page appears.
4. Enter the:
 - Display name for the database. The name must be unique among all scripted databases you are monitoring.
 - Description of the database (optional).
 - Host name. The name must be a valid name that can be resolved.
 - Host’s IP address (optional).
 - Name of and the full path to the database.

5. Select the **Enabled** check box to enable the resource monitor.
6. Click **Save**. The default monitoring plan page appears. You can modify the settings if you choose. The steps for editing scripted database monitoring plans are the same as for managed database monitoring plans. See the “[Monitoring managed databases](#)” section on page 1–5 for the procedure.

Until you start the monitoring agent, the database’s status appears as **Not Running**. You can use the available buttons to edit, copy, or delete the scripted database monitor from Fathom. For details about how to enable a scripted database agent, see the “[Starting a scripted database agent](#)” section on page 1–10.


Starting a scripted database agent

Before you can monitor a scripted database, you must start the Monitoring Agent.



To enable a scripted database agent:

1. From the **Database Details** page click **Control**, and copy the command line provided. For example, the following figure from the **Scripted Database** page shows the command line values to copy:



```
Command line to start Monitoring Agent
dbagent -db C:\OpenEdge\WRK\demo -H nightcap -S 8835
```

The command line includes the following information:

- The name and path of the scripted database (`-db <database-name>`).
 - The host machine on which Fathom is running (`-H <host-name>`).
 - The port on which Fathom is listening (`-S <port-number>`).
2. On the machine hosting the database, paste the command line at the **proenv** prompt. You can now access the scripted database’s control information and views.

Note: Using the command line supplied this way, add the start of your dbagent to your database start script. Each time your database starts up, the dbagent is automatically started and Fathom Management automatically starts monitoring your database.

Because the database is scripted, the following links normally available on a local database's details page do not appear:

- **Log File Monitor.**
- **Log File Viewer.**
- **File Systems View.**

Understanding and using database rules

Fathom supplies each managed database with a default rule set. Fathom also provides over 20 different database rules that you can define for a monitoring plan. These rules allow you to customize your database monitoring plans so that Fathom monitors your databases for specific conditions.

Figure 1–2 shows the **Available Rules** page.



Figure 1–2: Available Rules page

Only the rules that are not already part of the monitoring plan appear in the list. When you select any of these rules, the specific criteria associated with each rule appear. You can modify the default values associated with each individual rule.



To see a list of rules associated with a monitoring plan, click **Edit** on the **Monitoring Plan** page. The **Edit Monitoring Plan** page appears.

For each rule shown in the **Rules selected for this plan** section, the following details are provided:

- A colored dot, preceding the rule name, that indicates the status associated with each rule. See the *Resource Monitoring Guide* for general resource status information.
- The status represented by the colored dot.
- The alert severity for each rule that has failed.

Table 1–2 lists possible rule status values.

Table 1–2: Resource status legend

Status	Dot color	Description
Pass	Green	The resource monitor is currently working.
Fail	Red	<p>The most recent test involving the associated resource failed. This includes statuses such as tardy, timeout, unreachable, and so forth. Check the Alert Summary page or the specific monitor for possible alert details.</p> <p>This status can also identify an internal error that prevents the resource from being monitored.</p>
Not checked	Yellow	The resource monitor's status is currently unknown. For example, if system startup has just occurred, it is possible that the resource has not yet been polled.
Not running	Blue	The resource is currently not running. Watch resources such as databases and servers for this status, as they must be running before you can monitor them.
Disabled	Dark gray	The resource monitor has been disabled and is not currently monitoring a resource.
Inactive	White	There is no active monitoring plan.
Offline	Light gray	The resource is currently offline.

Working with database rule sets

A *database rule set* is a set of rules that you associate with one or more database resources through a monitoring plan. Database rule sets are stored in the **Fathom Component Library** and provide a way for you to manage many databases by sharing rule definitions. In this way, you create a common set of rules that you can set up for multiple databases.

Because each database rule set you create is stored in the **Fathom Component Library**, the rule set is available for your reuse or for use by others.

You can also add individual rules to a monitoring plan, whether or not the rules are part of any rule set. Note that if you include a rule in a monitoring plan's rule set and then add the same rule individually again with modifications, the rule in the rule set is overridden by the rule with the modifications.

Fathom provides a default database rule set. When a database resource is added to Fathom, a default monitoring plan, with the default rule set, is assigned to it.

Creating a database rule set

When associating rules sets with database resources, you can use the default rule set provided by Fathom or create your own.



To create a database rule set:

1. From the **Fathom Management console menu bar**, click **Library**. The **Fathom Component Library** page appears.
2. Click **Create Database Rule Set**. The **Create Database Rule Set** page appears.
3. In the **Name** field, enter the name of the rule set (no spaces allowed).
4. In the **Description** field, enter a brief description of the rule set.
5. Click **Save**. The **Database Rule Set** page appears. The rule set now appears in the list frame under **Rule Sets→Database**.
6. Click **Add Rule** to select the rules you want in the set. The steps for adding rules to a set are the same for adding rules to a monitoring plan. See the [“Modifying database monitoring plans”](#) section on page 1–7 for information on adding rules.

Note the following about rule sets:

- Once you create a rule set, you can edit, copy, or delete it.
- If you add a rule or a rule set to an existing rule set, the change affects all databases using the rule set.

Editing a database rule set

You can change the name or description of a rule set.



To edit a rule set, click **Edit** on the **Database Rule Set** page.

Note: You can access the list of existing database rule sets at any time from the **Fathom Component Library** list frame.

Copying a database rule set

You can copy a database rule set and make whatever modifications you want. You must, at a minimum, be sure to give the copy a unique name.



To copy a database rule set:

1. From the **Database Rule Set** page, click **Copy**. The **Copy Database Rule Set** page appears.
2. Provide a new name for the copy and (optionally) change the description.
3. Click **Save**.

You can now add one or more rules to the copy from either the **Copy Database Rule Set** page or the **Database Rule Set** page.

Deleting a database rule set

You can delete a rule set as long as it is not currently associated with any resource monitoring plans.



To delete a rule set:

1. From the **Database Rule Set** page, click **Delete**.
2. Click **OK** to confirm the deletion.

Rule sets with one or more rules in common

If you have multiple rule sets associated with a monitoring plan and you edit one of the rule sets, evaluation of only the first occurrence of any identically named rules takes place when the resource is polled. Which occurrence is considered “first” is determined by the alphabetic order of the rule set.

Associating a rule set with a database monitoring plan

You create a database rule set to associate and use it with one or more monitoring plans. Once you establish the association, the rule set is active for the database whenever the monitoring plan is active.



To associate a database rule set with a database monitoring plan:

1. Click **Resources** in the Fathom Management console. Expand the entries in the list frame to list your database resources.
2. Click the name of the database for which you want to access information. The **Database Details** page appears.
3. Click **Monitoring Plans**. The **Monitoring Plans** page appears.
4. Click the monitoring plan you want to update.
5. Click **Edit**. The **Edit Monitoring Plan** page appears.

6. Under **Rules selected for this plan**, click **Select Rule Sets**. A list of available rule sets appears.
7. Select one or more rule sets you want to associate with the plan.
8. Click **Save** when you finish. The monitoring plan is updated, and the **Monitoring Plan Definition** page appears.

Using the Configuration Advisor

The Configuration Advisor calculates suggested threshold values for rules by analyzing trend data. When you apply a recommended rule threshold setting, the alerts triggered as a result of rule violations provide a more meaningful indication of your resource's performance than if you were to arbitrarily set rule values.

The Configuration Advisor analyzes a rule's past performance for a specified period of time and, based on this data, indicates a baseline value. A *baseline value* is a number that serves as the base for calculating a set of possible threshold settings based on your system's past activity for a specific rule.

It is recommended that you use the Configuration Advisor-related default values for a set period of time (for example, one week) to capture data to the FathomTrendDatabase for a rule. This initial step will provide you sufficient data with which to perform the comparison.

Keep the following points in mind before you run the Configuration Advisor for the first time:

- Each rule you want to analyze must have at least 32 data samples stored in the FathomTrendDatabase. This sampling provides sufficient data from which the Configuration Advisor can determine a baseline value and subsequently perform a successful analysis of each rule's data.
- A polled rule must currently be associated with an active monitoring plan for it to be a candidate for the Configuration Advisor to process.
- If your monitoring plan contains After-imaging (AI) rules but you do not have After-imaging enabled for the database, do not select AI rules for analysis. When a monitoring plan contains AI rules and AI is not enabled for the database, the value stored in the FathomTrendDatabase is zero. The Configuration Advisor works by contrasting the FathomTrendDatabase data against the rule's range of values. Since zero falls outside the range of values, the Configuration Advisor's analysis becomes meaningless and it returns a baseline value that is meaningless.

**To initiate the Configuration Advisor:**

1. Navigate to the desired **Database Details** page.
2. Click **Configuration Advisor**. The **Configuration Advisor** page appears:

Configuration Advisor
skye.FathomTrendDatabase

Submit Cancel

For rule threshold calculations to take place, data must be retrieved from the FathomTrendDatabase. Please select the start and end date range to use for calculating the thresholds.

Start Date: 2003 May 15
End date: 2003 August 21

Choose time period to analyze :

☐ Sun ☒ Mon ☒ Tue ☒ Wed ☒ Thu ☒ Fri ☐ Sat

☒ Full day (24 hours)

☐ From: 9 : 00 AM To: 5 : 00 PM

Select rules (for analysis):

☒ Buffer IO High
☒ Buffers Flushed at Checkpoint High
☒ Database Commits Low
☒ Physical Reads High

3. In the **Start Date** and **End date** fields, define a date range that Fathom will use to collect data from the FathomTrendDatabase. (The default data range covers one week.)
4. In the **Choose time period to analyze** field, identify the time frame that defines a representative period of time in which the rules are generally active, or being used; this time frame is the period against which you want Fathom to calculate your baseline activity. (The default time period is Monday through Friday, 9:00 AM to 5 PM.)

Note: Your monitoring plan schedules are not necessarily the best choice for a time frame. A schedule defines a period of time in which rules are in effect; it does not focus on time periods in which your resource usage is highest.

For example, you may use the default schedule (24 hours a day, 7 days a week) to monitor your system, but want to select Monday through Friday from 8:00 AM to 6:00 PM for calculating your baseline settings.

5. In the **Select rules (for analysis)** field, deselect any rules you do not wish to analyze. The Configuration Advisor can analyze the following database resource rules:

- AIW Writes Percent Low.
- BIW Writes Percent Low.
- Buffer IO High.
- Buffers Flushed at Checkpoint High.
- Busy AI Buffer Waits High.
- Busy BI Buffer Waits High.
- Checkpoint Length Short.
- Database Commits Low.
- Empty AI Buffer Waits High.
- Empty BI Buffer Waits High.


Only rules associated with a monitoring plan appear in the **Select rules (for analysis)** list.

Note: If you added AI rules to a monitoring plan but do not have AI enabled on your database, do not select the AI rules for analysis.

6. Click **Submit**.

Note: Once you click **Submit**, you can browse to other Fathom pages and perform other tasks. You can return to the Configuration Advisor at some later point to check status and result details.

As the Configuration Advisor calculates the threshold settings, the following information appears, reporting the progress of each calculation it is performing:



Configuration Advisor

skye.FathomTrendDatabase

Buffer IO High

finished

Buffers Flushed at Checkpoint High


working

Database Commits Low

Physical Reads High

Depending upon the criteria that you set on the initial **Configuration Advisor** page, the number of rules you selected, and other factors such as your machine’s speed, this calculation process could take some time.

When the Configuration Advisor completes its calculations, the following page appears:



Configuration Advisor

skye.FathomTrendDatabase

Update Selected Rules

Cancel

The following time period was used for analysis

Start Date: 2003/4/15

End Date: 2003/7/21

Days:

Sun Mon Tue Wed Thu Fri Sat

Hours:

From: 9:00 To: 17:00

Rule	Recommend Values	Weekdays		Weekends		Detail
		Select:	Update	Select:	Update	
Buffer IO High	13.5 [0]* Blocks read per block written	<div>All</div> <div>None</div>	<input checked="" type="checkbox"/> 75	<div>All</div> <div>None</div>	<input checked="" type="checkbox"/> 75	<div>Detail</div>
Buffers Flushed at Checkpoint High	4.0 [0]* buffers per checkpoint	<input checked="" type="checkbox"/>	15	<input checked="" type="checkbox"/>	15	<div>Detail</div>
Database Commits Low	4.0 [858]* commits per second	<input checked="" type="checkbox"/>	30	<input checked="" type="checkbox"/>	30	<div>Detail</div>
Physical Reads High	4.0 [0]* percent	<input checked="" type="checkbox"/>	5	<input checked="" type="checkbox"/>	5	<div>Detail</div>

The following time period was used for analysis section of this page summarizes the values defined on the initial Configuration Advisor page. These values are displayed here to help you to easily recall the time period criteria you set.

The **Rule** section contains all the rule-related calculated data. You can now review the Configuration Advisor's recommendations and choose whether or not to update your threshold values. See the [“Understanding the recommended threshold settings”](#) section on page 1–21 for more information about the Configuration Advisor's calculated values.

Note: The Configuration Advisor page displays the calculated results data until either **Update Selected Rules** or **Cancel** is clicked.

Understanding the recommended threshold settings

When you view the Configuration Advisor's analysis, each analyzed rule appears as an individual line item in the **Rule** section. Associated with each rule is a **Recommended Values** drop-down list; it contains one of the following entries:

- Numeric values that identify the recommended rule threshold settings. This list can contain up to several different numeric items; collectively, these values comprise the range of recommended threshold settings.
- An **Insufficient data for analysis** message. The Configuration Advisor displays this message when the criteria to perform the data analysis successfully is not met.

Reviewing recommended values

The Configuration Advisor displays a range of possible values from which to select. The initial value that appears in the **Recommended Values** field indicates the primary recommended threshold setting, based on the data analysis process.

Each recommended value is expressed as a set of two numbers. The left-most number specifies the recommended threshold setting; the right-most number, displayed in brackets, identifies the number of times this threshold value, if used with the collected data, would have broken the rule and triggered an alert. As you review the recommended threshold settings, keep in mind the rule behavior and alert notification frequency you want to establish for a resource.

Using the Detail button

Each row has an associated **Detail** button that displays information about the rule's analysis. The **Detail** page for a rule for which there is insufficient data for analysis identifies the number of samples found. This number is always lower than the minimum of 32 data samples required. Review this data to help you decide if you need to expand your time period to try and capture more samples and re-run the Configuration Advisor for a given rule.

Comparing and selecting threshold settings

By default, the Configuration Advisor assumes that you are going to select and submit one of the recommended threshold settings; it provides a check mark in the **Update** field for each rule row. However, you have options concerning the selection process. As you compare the existing and recommended values, you can elect to change none, some, or all values for a rule and each individual monitoring plan.

This section describes the procedure to compare the current rule setting with the recommended threshold settings, and to update each schedule with your specific selections. Perform this comparison to help you further determine your final selection.



To compare and select threshold settings:

1. For a specific row, note the value that displays in the **Current Threshold** field under a specific schedule.
2. Click **Recommended Values** to display the range of recommended values for the associated rule.
3. Compare the possible **Recommended Values** that display with the value in the **Current Threshold** field. As you determine the best threshold rule setting, keep your goals for this rule in mind. Also, consider any additional selection criteria as you compare values.
4. Repeat these steps for each rule and its associated monitoring plan.

Additional selection criteria

Reasons for selecting one value and not another can include:

- How often you want alerts generated.
- Factors unique to your resource's performance that you want to consider in making your selection.
- Knowledge of your system's operational needs and goals.

Submitting your threshold setting selections

When you click **Update Selected Rules**, Fathom applies all of your selections at the same time. There is no undo option associated with this group submission; to reset any values back to a previously defined setting, you must access the resource's monitoring plan, display the individual rule, and override the displayed value.

Note: The Configuration Advisor page displays the calculated results data until either **Update Selected Rules** or **Cancel** is clicked.

Determining the effectiveness of your selections

The most effective way to determine if your threshold adjustments are serving your needs is to review your alert notifications. Strive for a threshold setting that is consistent with your resource and business needs. If you receive alerts too frequently or infrequently to suit your operational needs, you might want to further refine your threshold settings.

Graphical Displays of Database Data

Fathom Management provides a graphical display of database data in several different areas of the product. For example, many of the database views on the **Database Details** page include graphical and text interpretations. Graphical displays of data allow you to check the status of your database resources at a glance. This chapter explains Fathom's graphical display of data, as described in the following sections:

- [Interpreting graphical database data](#)
- [General graph information](#)
- [Display characteristics of a pinup graph](#)

Interpreting graphical database data

Several of the database views provided by Fathom include both a text-based and a graphical interpretation of data. In some cases, the text-based data is accompanied by a corresponding graphical interpretation in the form of a pie graph, bar graph, stacked column graph, or meter graph.

Each graph includes a legend that explains what the different graph colors you see in the browser signify, as shown in the pie graph in [Figure 2–1](#).

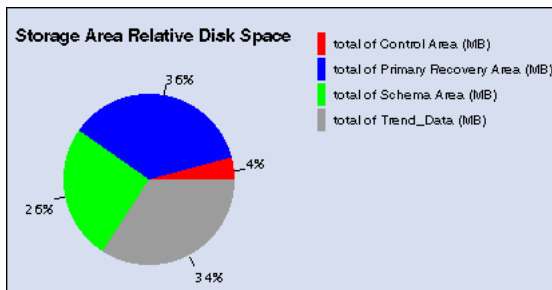


Figure 2–1: Graph legend

In [Figure 2–1](#), the legend accompanying the pie graph indicates that:

- The red section of the pie corresponds to the percentage of the disk space used by the Control Area.
- The blue section of the pie corresponds to the percentage of the disk space used by the Primary Recovery Area.
- The green section of the pie corresponds to the percentage of the disk space used by the Schema Area.
- The gray section of the pie corresponds to the percentage of disk space used by trended data.

Note: To see the above figure in color, refer to the online version of this guide.

Types of graphs

Graphs appearing in Fathom database views belong to one of three families: current, meter, or historical.

Current

Graphs that belong to the current family include: pie, bar, and column. Current graphs display the most current data samples.

Meter

Graphs that belong to the meter family include: full, half left, half right, half bottom, and half top. Meter graphs display the most current datum sample.

Figure 2–2 shows a full meter graph.

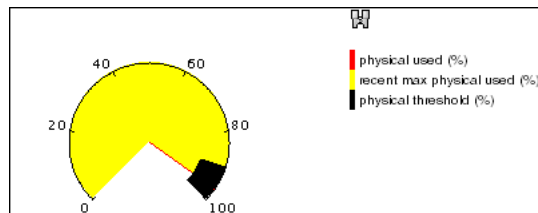


Figure 2–2: Full meter graph

The yellow background indicates the high water mark, while the red arrow indicates the current value. The black area of the graph indicates that a rule has defined a threshold for the data. The meter graph in Figure 2–2 shows that the resource has broken the rule’s threshold value.

Historical

Graphs that belong to the historical family include: line, hi/low, column, area, stacked column, and stacked area. Historical graphs show values over a period of time.

Hi/low graphs present the start, high, low, and end values of your data. See [Figure 2–3](#) for a sample hi/low graph.

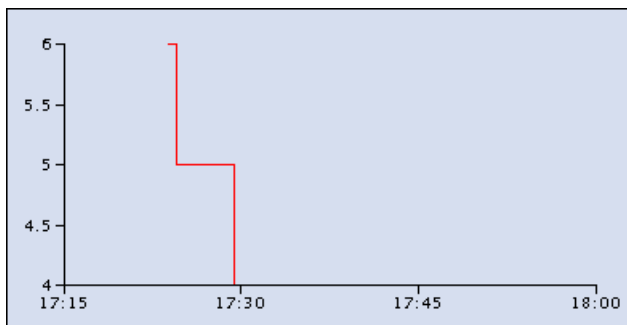


Figure 2–3: Hi/low graph

The highest vertical point on a hi/low graph indicates the high water mark (HWM). The lowest vertical point indicates the low water mark (LWM).

Historical graphs use dashed lines to indicate thresholds, as shown in [Figure 2–4](#).

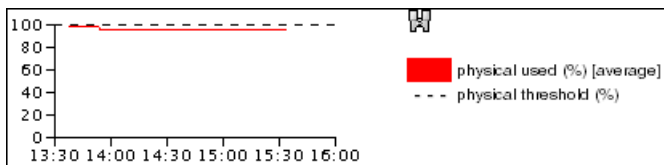


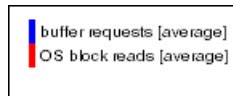
Figure 2–4: Graph with threshold indicated

General graph information

Keep the following information in mind when you are looking at graphs:

- The production of graphs is CPU-intensive. If you are monitoring CPU usage, an alert might fire when the graph is generated. To avoid firing such an alert, increase the number of failed polls after which Fathom throws an alert.
- Depending on the browser in which you are viewing a graph, the graph type and its property settings, and the number of data points displayed, you can display pop-up content details from within the graph. Review pop-up content to inspect resource activity in greater detail.

- Only those resources that have monitors defined for them produce information that is graphed.
- A yellow border around a viewlet graph on the **My Fathom** page indicates that the database is inactive or does not have a monitoring plan. A red border indicates that the container or database is offline. For more information about viewlets and the **My Fathom** page, see the [Resource Monitoring Guide](#).
- If data averaging is turned on for viewlet graphs on the **My Fathom** page, “[average]” appears next to the unit in the graph’s legend:



- Note that any discontinuity in the caching of a database resource’s data will be reflected in the graphs. Caching discontinuity occurs when a resource goes off line or a resource monitor is disabled. Discontinuities are obvious in line graphs, as they appear as a break in the line. Note that caching discontinuities are harder to distinguish on area or column graphs, since gaps in area or column graphs can indicate a zero value or caching discontinuity.
- All graph data is stored in `fathom-work-dir\cachedata\fathomdatacache.odt`. If Fathom is not running, you can delete the database, but doing so will cause you to lose all of your cached graph data.
- By default, 48 hours of graph data is stored for each database resource. You can set a different collection setting on the **Graph Cache Database Configure** page. You access the **Graph Cache Database Configuration** page from the **Configuration** section of the **Options** list frame. See the [Installation and Configuration Guide](#) for more details about setting the graph cache.

Display characteristics of a pinup graph

If a graph is accompanied by a small binocular icon, you can change its size, style, and how often it refreshes. You make these changes in the pinup version of the graph.



To create a pinup graph:

1. Click the binoculars icon in the lower right of the graph whose view you want to modify. The pinup graph window opens, displaying the graph.
2. Drag the lower-right corner of the window so you can see the entire page. The data label at the top of the graph serves as the graph’s legend.
3. From the pinup you can, depending on the graph, customize the graph properties described in [Table 2–1](#).

Table 2–1: Graph properties and options (1 of 3)

Property	Options	Comments
Graph size	<ul style="list-style-type: none">• Very small• Small• Medium• Large	If you have a graph with small statistics, you can choose to make the pinup graph larger so you can better see its details.
Graph style	Vary by graph family.	See the “ Types of graphs ” section on page 2–3 for a list of graphs that belong to each family.
Graph data averaging	<ul style="list-style-type: none">• On• Off	When data averaging is on, Fathom computes weighted averages from the data. Using weighted averages reduces the number of data points and creates more meaningful graphs. If data averaging is off and there is more data than pixels, Fathom compresses the data.
Graph dimension	<ul style="list-style-type: none">• 2D• 3D	Changes the display from two dimensional to three dimensional.

Table 2–1: Graph properties and options*(2 of 3)*

Property	Options	Comments
Grid	<ul style="list-style-type: none">• On• Off	Default is Off.
Graph max time	<ul style="list-style-type: none">• 5 minutes• 10 minutes• 15 minutes• 20 minutes• 30 minutes• 45 minutes• 1 hour• 2 hours• 4 hours• 8 hours• 12 hours• 24 hours• 36 hours• 2 days	Controls how much time the graph spans. Note: The graph time span does not affect how often or how much data is collected for graphing. You change how often your database resources are polled and how often the polled data is trended by editing the resource's monitoring plan. See Chapter 1, "Database Resource Monitoring" for information about editing monitoring plans.

Table 2–1: Graph properties and options (3 of 3)

Property	Options	Comments
Graph start time	--	Select the check box and choose the start time from the drop down menus. Note: The time selected does not necessarily correspond to the first time that displays on the graph axis. For example, if you define the pinup graph to display data for 15 minutes starting at 10:05 AM, the axis may run from 10:00 AM to 10:30 AM.
Refresh rate	<ul style="list-style-type: none">• None• 15 seconds• 30 seconds• 1 minute• 2 minutes• 3 minutes• 4 minutes• 5 minutes• 10 minutes• 15 minutes	The refresh rate is the rate at which the resource is checked to see if there is more information to put in the graph. The refresh rate should not be less than the polling rate for the resource. For example, if you set the refresh rate to 1 minute and the polling rate is at 5 minutes, you do not get new graph data every minute; you get it only at the same rate as the polling occurs.

4. Click **Change Pinup** when you finish making your selections. The graph appears in the pinup with the new characteristics.

Note the following about pinup graphs:

- You cannot save the pinup graph settings.
- The database view page is not updated to reflect the graph characteristics you chose in the pinup.

Examining Data from an OpenEdge Database

Your database is one of the most critical operational resources you maintain at your site. Fathom Management provides you with several ways to view and analyze data collected from your database resources so you can better manage them. This chapter contains the following sections:

- [Accessing database information](#)
- [Changing database control settings](#)
- [Database log file monitors](#)
- [Using the log file viewer](#)
- [Understanding OpenEdge database views](#)

Accessing database information

The **Database Details** page, shown below in [Figure 3–1](#), provides access to detailed information about an individual database.



Figure 3–1: Database Details page



To access the Database Details page:

1. Click **Resources** in the Fathom Management console menu bar. The main resource types appear in the list frame.
2. In the list frame, browse to and click the desired database. The **Database Details** page for your database appears in the detail frame.

A brief description of each link on the **Database Details** page appears below the link itself. The links on the page are enabled as follows:

- The **Control** and the **Log File Monitor** links are enabled when you access this page.

Note: **Log File Monitor** links do not appear on scripted databases' **Database Details** pages.

- When the database and the database agent are running, as shown in the **Database status** section of the page, all links are enabled.

The following information is also available in the upper-right corner:

- The database's current status, and how long it has been at that status.
- The date and time of the last poll.
- The number of polls taken.
- The number and percent of failed polls.

Click any of the links on the page to access relevant details about the database.

Changing database control settings

From the **Database Details** page you can access the **Control** page. The **Control** page provides information about the current database and database agent settings and allows you to start and stop a managed database and agent. For a scripted database, the page allows you to see the database status and the remote monitoring agent status, and to copy the command line that you use to start the dbagent on the remote machine. You can also stop the remote monitoring agent from a scripted database's **Control** page.

Figure 3–2 shows a sample **Database Control** page for a managed database. Figure 3–3 shows a sample **Database Control** page for a scripted database.

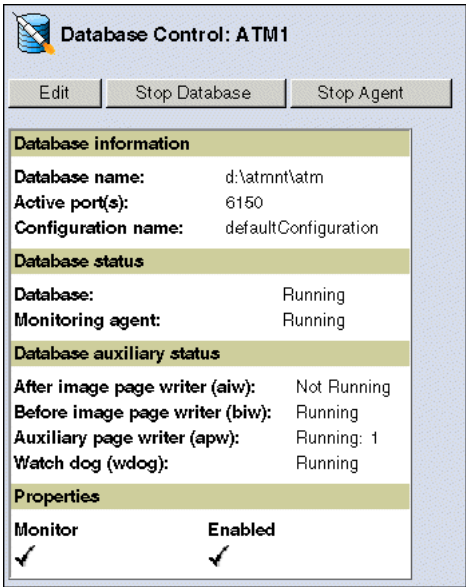


Figure 3–2: Managed Database Control page

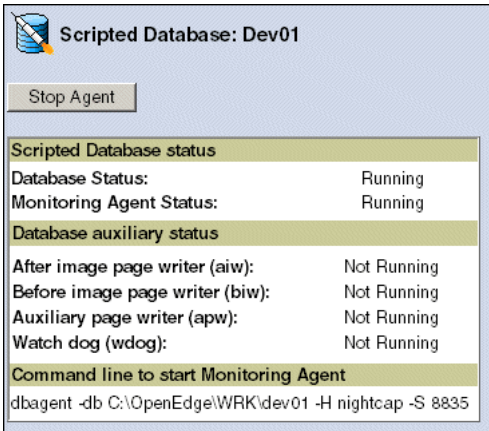


Figure 3–3: Scripted Database Control page

From the **Control** page of a managed database, you can:

- Stop the database by clicking **Stop Database**; the status is updated to **Not Running**. To restart the database, click **Start Database**. (This is the same button; the button's text changes to display the opposite status of the one that is currently set.)

Note: The database agent, or monitoring agent, works only if the database is running. Therefore, when you stop a database, the monitoring agent status automatically changes to **Not Running**. However, when you start the local database, the agent does not automatically start.

- Change the status of a database monitor from **Enabled** to **Disabled** by clicking **Edit** and clearing the **Enabled** check box. If you disable database monitors, the database still appears under **Databases** in the list frame. However, the database is not being monitored by Fathom.

Consider the following points as they relate to deleting databases rather than disabling them:

- If you delete a database from within Fathom, you have permanently removed it from Fathom. It is preferable to disable the database because you have more options to use the resource at a later time. For example, you could elect to copy and reuse the database's configuration for a new database you create, or you could migrate the database using the Database Migration Utility. Alternatively, you can delete it and add it again in the future if necessary.
- Similarly, if you delete a managed database through the Progress Explorer, the database's status within Fathom, by default, becomes scripted, and the database is disabled. As with a database that you directly disable from within Fathom, a disabled database continues to display in the Fathom Management console list frame with a disabled status circle preceding it. It is not permanently removed from Fathom. You can elect to copy the configuration at a later date, migrate the database using the Database Migration Utility, or delete it in Fathom.

Note that you can still continue to run historical reports against a database after you have deleted it from Fathom.

Database log file monitors

Fathom Management automatically creates a *database log file monitor* for each managed database started through Fathom or the Progress Explorer.

The database log file monitor can help you:

- Ensure the integrity of database log files by monitoring files for errors and allowing you to define actions to trigger when errors occur.
- Use predefined database-related search criteria, or create your own, to run against the data in your database file. You create and maintain search criteria in the Fathom Component Library.

Specifically, the predefined search criteria provide detailed data about the recorded operations of your database and provide you a means to extract this data. This data can help you effectively respond to and manage your database.

Note: You can also create a log file monitor for use with files such as log files, text files, or any non-OpenEdge log file. The search criteria feature works the same in each of these Fathom resource types. For information about the log file monitor and detailed information about search criteria that pertain to both of these resource types, see the [Resource Monitoring Guide](#).

Log file monitor default values

The predefined database log file monitor that Fathom creates for each database contains several default values. Of the default database log file monitor properties, you can modify only the **Enabled** property and its description. See the “[Customizing a database log file monitor](#)” section on page 3–7 for details on creating a customized log file monitor.

The default values are as follows:

- The **Bookmark** is set to **Last Line**, and it is unique.
- The database default log file monitor is disabled until the database is first started.
- The **On First Poll** property is set to **Search From End**.

Log file monitor considerations

Review the following considerations before you customize a database log file monitor:

- The database log file monitor is not enabled until the database for which it was created is started. When the database log file monitor first starts monitoring a database, it always starts at the end of the log file.
- The database log file provides predefined search criteria that address common database-related events. You can use these searches as defined, or copy and customize them.
- You can also create your own search criteria. For example, if there is a particular database error for which you want to monitor a database, you can use the promsg number as the search text.
- The FathomTrendDatabase is a managed, OpenEdge database. Therefore, the FathomTrendDatabase has a predefined database log file monitor that you can customize to meet your needs.

Customizing a database log file monitor

The database log file monitor that Fathom creates for each database contains default values. You can use these values or customize the log file monitor to better suit your needs.

Note: This guide presents the basic steps needed to customize a log file monitor and does not go into detail about search criteria. See the [Resource Monitoring Guide](#) for detailed information about creating and changing search criteria.



To customize a database log file monitor:

1. From the **Database Details** page of the desired database, click **Log File Monitor**. The **Log File** page appears.
2. Decide if you will customize the existing plan or add a new one.
 - Click **Add Plan** to add a monitoring plan to this resource monitor.

Note: Fathom prevents the assignment of schedules that share days or have times that overlap. For example, if you have a `Default_Schedule` set up for a resource monitor, you cannot set up an additional plan because the `Default_Schedule` is defined for 7 days a week, 24 hours a day. You must modify or remove the `Default_Schedule` to set up additional plans.

- To customize the existing plan, click **Edit** for the plan you want to customize. The On the **Edit** page, change the polling interval, if desired. Select the **Alerts Enabled** check box to enable alerts.
3. From the **Rules selected for this plan** section, click either **Add Rule** or **Select Rule Sets**.
 4. After you either add a rule or select from the list of predefined rule sets, click **Save** to return to the monitoring plan detail page.

Using the log file viewer


The log file viewer allows you to examine a database log file through an HTML interface. This section describes viewing database log files.

Table 3–1 lists where information about non-database log files is found.

Table 3–1: Documentation about log files

For information about. . .	See the . . .
AdminServer (Fathom) log files	Reporting Guide
AppServer log files	OpenEdge Server Management Guide
NameServer log files	OpenEdge Server Management Guide
WebSpeed log files	OpenEdge Server Management Guide

► To access the log file viewer, click **Log File Viewer** in the **Command and Control** section of the **Database Details** page. The viewer appears:


Database: Sports2000
Log File Contents
May 25, 2004 12:14:57 PM

First Prior Next Last Reload
Go To
Show: Overlap:
Display: ☒ Ascending ☐ Descending

Sports2000 log file summary
Size of log: 824.99 KB
Lines in log: 13,160
Display start line: 13,141
% of log at start line: 99.9 %
Log file status: unchanged

```

13,141 09:48:23 BROKER 0: Number of After-Image Buffers (-aibufs): 5. (4256)
13,142 09:48:23 BROKER 0: Storage object cache size (-omsz): 1024 (8527)
13,143 09:48:23 BROKER 0: Maximum Number of Clients Per Server (-Ma): 5. (4257)
13,144 09:48:23 BROKER 0: Maximum Number of Servers (-Mn): 5. (4258)
13,145 09:48:23 BROKER 0: Minimum Clients Per Server (-Mi): 1. (4259)
13,146 09:48:23 BROKER 0: Maximum Number of Users (-n): 21. (4260)
13,147 09:48:23 BROKER 0: Host Name (-H): NBELETOURN. (4261)
13,148 09:48:23 BROKER 0: Service Name (-S): 2000. (4262)
13,149 09:48:23 BROKER 0: Network Type (-N): TCP. (4263)
13,150 09:48:23 BROKER 0: Character Set (-cpinternal): ISO8859-1. (4264)
13,151 09:48:23 BROKER 0: Parameter File: Not Enabled. (4282)
13,152 09:48:23 BROKER 0: Maximum Servers Per Broker (-Mpb): 4. (5647)
13,153 09:48:23 BROKER 0: Minimum Port for Auto Servers (-minport): 3000. (5648)
13,154 09:48:23 BROKER 0: Maximum Port for Auto Servers (-maxport): 5000. (5649)
13,155 09:48:23 BROKER 0: This broker supports both 4GL and SQL server groups. (8865)
13,156 09:48:23 BROKER 0: Created shared memory with segment_id: 10092544 (9336)
13,157 09:48:36 WDOG 5: Started. (2518)
13,158 09:52:44 FMAGNT 6: Login by SYSTEM on CON:. (452)
13,159 09:52:44 FMAGNT 6: Usr 6 set name to DB_Agent. (7129)

```

The following information will help you use the Log File Viewer:

- Use the **Show** field to control how many database log file entries display at one time. The number entered into the **Show** field cannot be less than 10.
- Use the **Overlap** field to control how many entries are repeated from screen to screen.

Note: The value in the **Overlap** field cannot be more than the number in the **Show** field minus one. For example, if you show 30 entries, you can overlap only 29 or fewer of them.

- Click **Reload** after changing the values in the **Show** field or **Overlap** field. If you do not reload, the viewer displays the previous values.
- Click **Go To** to control which numbered entry in the log file the viewer begins its display with. For example, a value of 10 entered into the **Go To** field will begin the display from the tenth log file entry.

Note: You must click **Go To** after entering a value in the **Go To** field or the viewer will not update its display.

- The default display of entries is in ascending order; choose **Descending** to change the display. Note that the **Show** field dictates the number of entries shown, whether they display in ascending or descending order.
- Click **First** to display the first x entries, where x is the value in the **Show** field.
- Click **Previous** to display the previous x entries, where x is the value in the **Show** field.
- Click **Next** to display the next x entries, where x is the value in the **Show** field.
- Click **Last** to display the last x entries, where x is the value in the **Show** field.
- To view additional log file entries without changing your current starting log file entry, leave the **Go To** field blank, change the value in the **Show** field, and click **Reload**.
- If the contents of the log file have changed since you opened the viewer, the log file viewer indicates this in the **Log file status** field.

- Fathom considers a viewer that has been inactive for more than four hours “stale.” Once a viewer becomes stale, Fathom releases ninety-five percent of any memory it holds. If you try to use a stale viewer, Fathom automatically reloads the file. Because additional resource activity might have occurred during the viewer’s inactivity, the reloaded log file view might not match the previous log file view of that resource.
- Fathom considers a viewer that has been inactive for forty-eight hours “dead.” Once a viewer dies, Fathom releases all of its memory. To return to the log file displayed in a dead view, you need to renavigate to it, even if you pinned up the view or saved a link to it before the viewer died.

Understanding OpenEdge database views

The various database views display data about key database components so that you can assess what is happening with the databases. The database agent gathers data for the views by polling the database’s Virtual System Tables (VSTs). (View data does not come from the FathomTrendDatabase.) For more information on VSTs, see *OpenEdge Data Management: Database Administration*.

Views capture information at a specific point in time. Because database status can change from moment to moment, refresh the detail frame to keep the view’s data current. All views display the date and time they were built.

Different database views appear in different formats. Some include graphical displays of data, such as pie graphs or column graphs (for more information about graphical displays of data, see [Chapter 2, “Graphical Displays of Database Data”](#)). Views created for scripted databases do not contain graphical displays of data.

Some views allow you to narrow the scope of data by clicking on any underlined subject. Note that underlined subjects behave differently than the underlined column headings. Clicking an underlined column heading does not bring you to a more detailed view of the subject; instead, it changes the column’s sort order. If the default sort order for a column is ascending, clicking the heading will change the sort to descending. If the default sort order is descending, clicking the heading will make it ascending. Note that when you change the sort order, the page refreshes.

Database views are organized into three overall categories: **Operations** views, **Informational** views, and **Advanced** views. [Table 3–2](#) lists and describes all database views, and indicates if the view can be created for both managed and scripted databases or for only managed databases.

Table 3–2: Database views

View name	Database type	Description
Vital Signs	Managed and scripted	Summary of important database information.
File Systems	Managed	List of file systems and associated files that the database uses. (Not available for remote databases.)
Storage Areas	Managed and scripted	Storage area and extent information and statistics.
User Activity	Managed and scripted	Information about connected users and user statistics.
Record and Index Activity	Managed and scripted	I/O activity for tables and indexes.
Locks and Latches	Managed and scripted	Information about lock and latch activity.
Transactions	Managed and scripted	Information about database transactions.
Memory Resources	Managed and scripted	Summary of buffers, memory, etc.
Page Writers	Managed and scripted	Summary of page writer activities.
Start Parameters	Managed and scripted	Detail about the database start parameters.
General Details	Managed and scripted	General information about the database: start, backup, etc.
Raw VST Data	Managed and scripted	Examination of individual VST tables.
Raw System Table Data	Managed and scripted	Examination of individual system tables.

Note: You access the above views from the **Database Details** page.

Vital Signs view

The **Vital Signs** view shows a page of information relevant to the critical operations of the database.

The data populating the **Vital Signs** view come from many VSTs, including _ActAILog, _ActPWLog, _ActBuffer, _ActSummary, _UserIO, and _ActIOType. [Figure 3–4](#) and [Figure 3–5](#) show a sample **Operations Status** view.

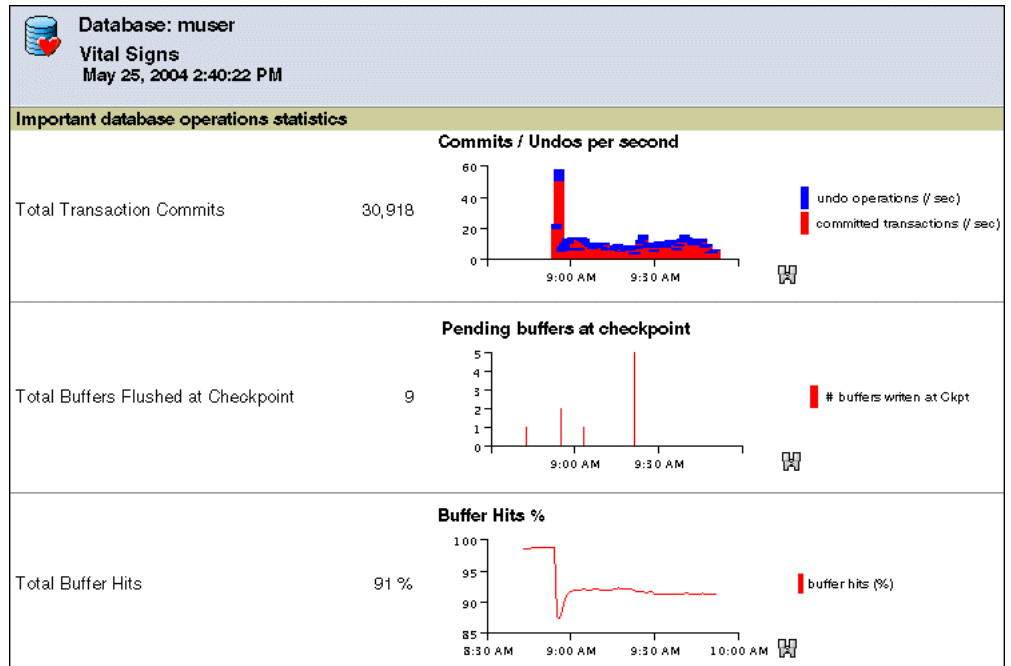


Figure 3–4: Vital Signs view page (1 of 2)

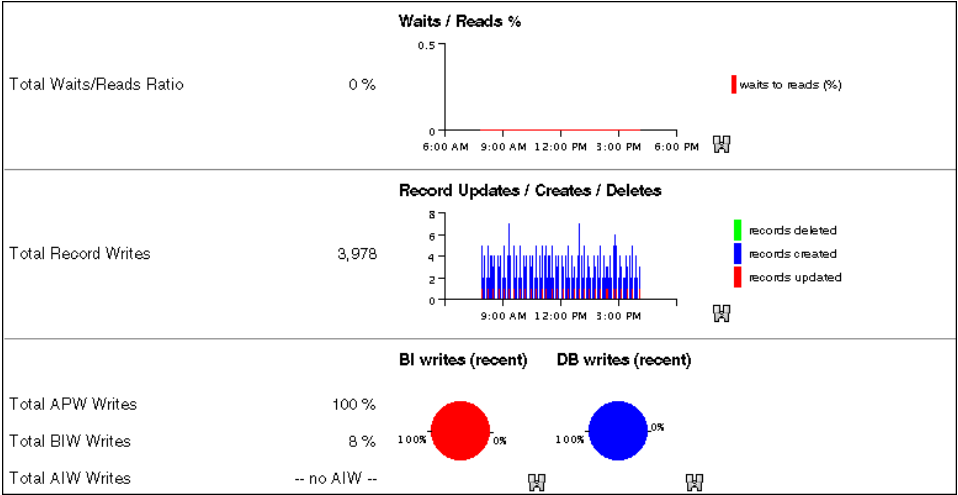


Figure 3–5: Vital Signs view page (2 of 2)

Note that the figures to the left of the operation statistics indicate totals since the database was started, while the graphs indicate activity over time.

File Systems view

Figure 3–6 and Figure 3–7 show the **File Systems** page that appears when you select the **File Systems** link from the **Database Details** page. This initial view displays a list of file systems and associated files on a local machine that a specific database uses.

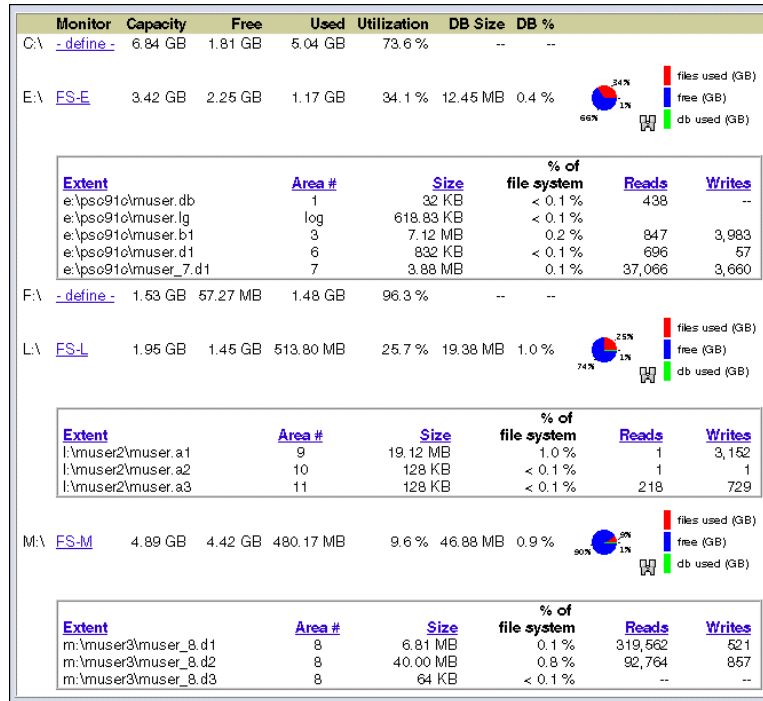


Figure 3–6: File Systems view (1 of 2)

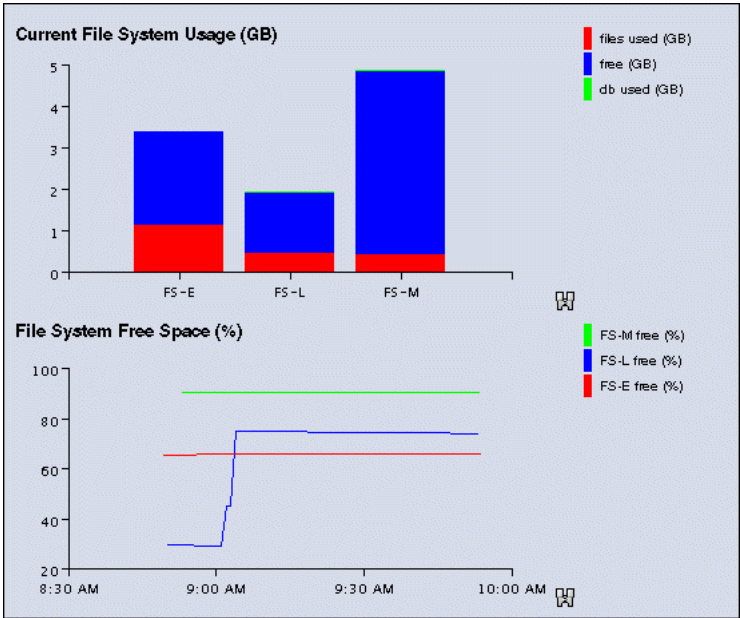


Figure 3-7: File Systems view (2 of 2)

The views present a snapshot of this data at the date and time specified at the top of the page. To avoid viewing stale data on this page, click the **Refresh** icon periodically to keep your view current.

The database extent data on the **File Systems** page originate primarily from the `_FileList` VST. Data appearing in the **Size** column originate in the `_AreaExtent` system VST.

Note: If you access the **File System** page from the **Fathom Resources** page instead of from the **Database Details** page, Fathom displays identical data. The only exception is that when you access the **File System** page from the **Fathom Resources** page, Fathom shows all databases for a system.

The **File Systems** page includes general file system information and detailed storage area information arranged under the following headings:

- **Monitor** — The name of a file system monitor. Only those file systems for which a monitor has been defined display a specific monitor name. Those file systems for which a file system monitor is not currently set up display the entry - **define** - under the **Monitor** heading. See the “[Creating file resource monitors](#)” section on page 3–17 for more information.
- **Capacity** — The byte capacity for each file system.
- **Free** — The number of bytes that are currently free.
- **Used** — The number of bytes currently used.
- **Utilization** — The number of bytes currently used, expressed as a percent of the capacity. Percent amounts display one decimal place of precision.
- **DB Size** — The size of all extent files on that file system in bytes.
- **DB%** — The DB size as a percent of the capacity.
- **Extent** — The filename and path of the database’s storage area extents.
- **Area #** — The database storage area number.
- **Size** — The size in bytes.
- **% of file system** — The size as a percentage of the file system capacity.
- **Reads and Writes** — The number of block read/writes that have occurred since the database was started.

Creating file resource monitors

You can set up or access an existing file resource monitor for any file system listed on the **File System** view page.



To set up a monitor, select **-define-** next to the file system. The **Create File Monitor Resource** page appears. Once you create a file system resource monitor, the resource name replaces the **-define-** label.

You create monitoring plans and rules for this resource monitor following the procedures outlined in the [Resource Monitoring Guide](#).

Storage Area views

The **Storage Area Utilizations** view allows you to view the status of your database storage areas. Information you find in the **Storage Area** view includes the percentage of use for each area, the size of each area, and an area's percentage of reads and writes relative to the monitored database's reads and writes.

Sample **Storage Area Utilizations** views are shown in [Figure 3–8](#), [Figure 3–9](#), and [Figure 3–10](#).

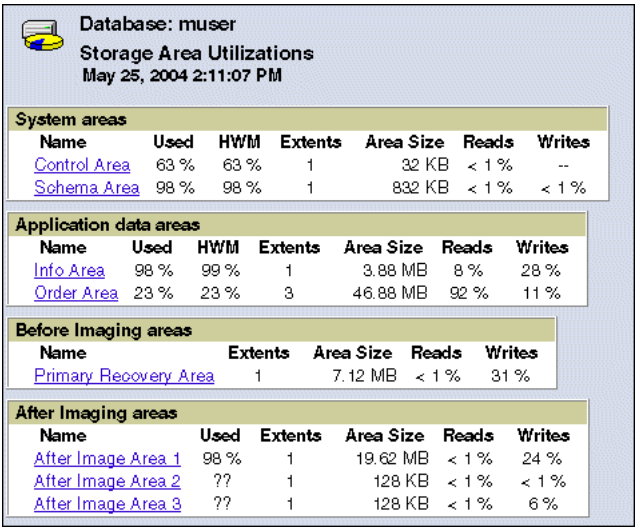


Figure 3–8: Storage Area view (1 of 3)

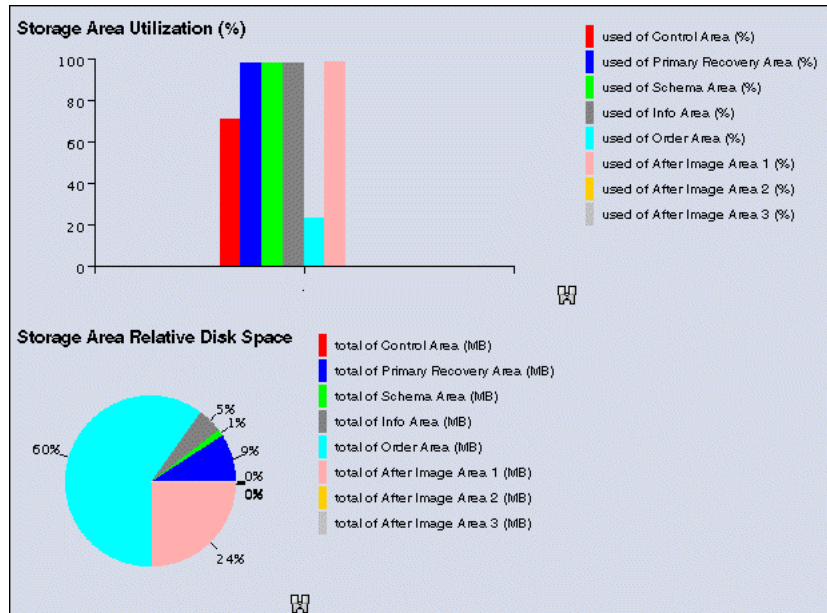


Figure 3–9: Storage Area view (2 of 3)

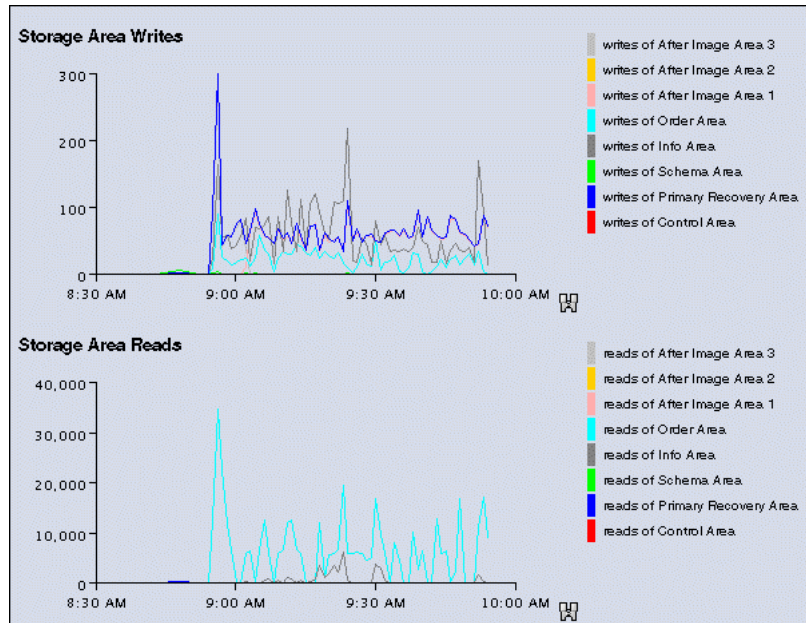


Figure 3–10: Storage Area view (3 of 3)

The following information will help you interpret the data presented on the **Storage Area Utilizations** page:

- Reads and writes are in database blocks.
- The percentage listed in the **Reads** column and **Writes** column is a percentage of the database's total reads or writes. For example, in [Figure 3–8](#), the percentage of writes for the Order area accounts for 11% of the database's entire writes.

The absence of an entry in the **Reads** column or **Writes** column indicates that no reads or writes have taken place.

- An entry of less than one percent in the **Reads** column or **Writes** column indicates that reads or writes took place, but not enough to reach one percent.
- The **Storage Area Utilization** graph indicates the fullness of every storage area, while the **Storage Area Relative Disk Space** graph indicates the relative percent of database space used by each storage area.
- The **High Water Mark (HWM)** column indicates the greatest number of blocks used in a storage area. The following equation produces the percentage displayed in the **HWM** column: x/y , where x is the number of the HWM block and y is the total number of blocks existing in the storage area. For example, if your database's HWM block is 5 and its total number of existing blocks in the storage area is 8, the data displayed in the **HWM** column would be 63% (or $5/8=.625$ rounded).
- The **Used** column indicates the amount of space used on a storage area's free chain. The following equation produces the percentage displayed in the **Used** column: $(x-z)/y$, where x is the number of the HWM block, z is the total number of free blocks in the area's extents, and y is the total number of blocks existing in the storage area.
- Double question marks in a data column indicate that the database agent could not access the needed information. See the [“After Imaging areas”](#) section on page 3–25 for more information about double question marks.
- Double exclamation points in a data column indicate that the table was unavailable at the time the **Storage Area Utilization** page was built. For example, if the database agent is down and therefore unable to poll the database at the time of the view's creation, double exclamation points appear in the data columns.

System areas

The **System Area** section contains information about the control and schema areas. Clicking **Control Area** or **Schema Area** at the top of the **Storage Area Utilizations** page takes you to a more in-depth view of each. [Figure 3–11](#) shows the **Control Area** page.

Note: Although it is possible to create application extents in the schema area, Fathom lists the schema area under the **System Areas** view. When best practices are followed and all application extents are placed in their own areas, the schema area stores only system information.

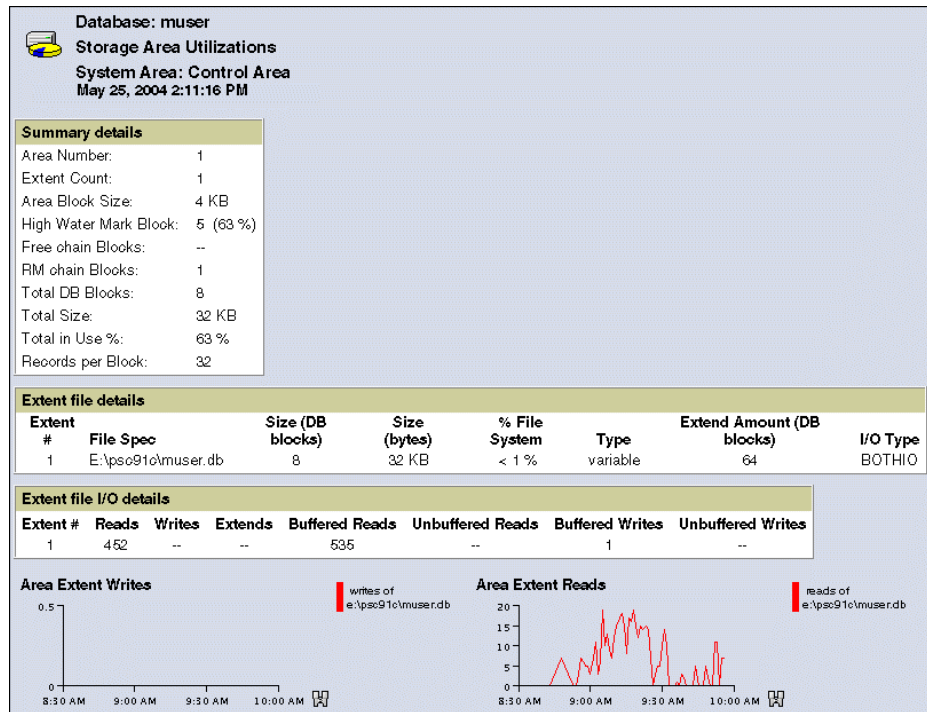


Figure 3–11: Control Area view

The data presented in the **Control Area** and **Schema Area** views come from the `_AreaStatus`, `_FileList`, and `_ActIOFile` VSTs.

Application data areas

Names of areas listed in the **Application Data** section will vary according to the database being viewed. Clicking on a listed application area brings you to a more detailed view of it. In [Figure 3–12](#), Order is the data area being viewed. [Figure 3–13](#) shows additional information displayed in the **Order Area** view.

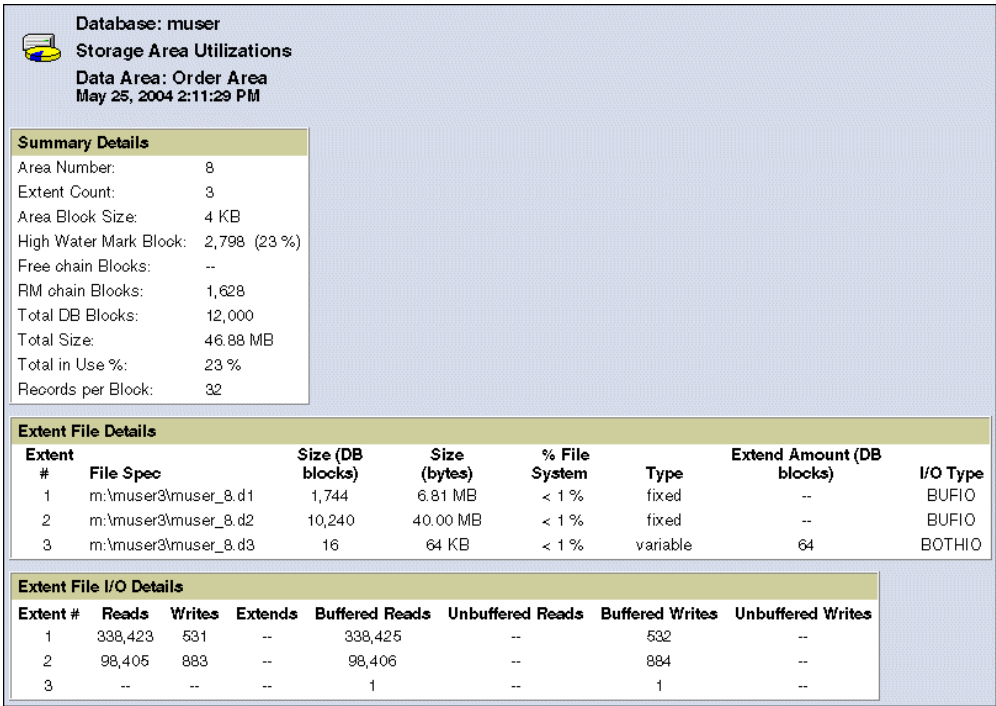


Figure 3–12: Order Area summary view

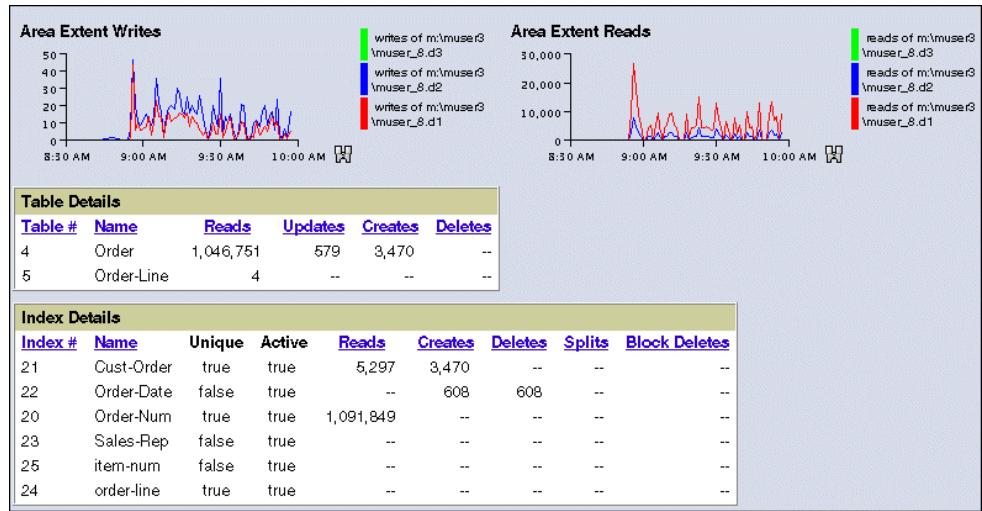


Figure 3-13: Additional Order Area view

The line charts shown in Figure 3-13 indicate reads and writes by extent.

Note: The statistical data displayed in the **Table** and **Index Details** sections are affected if the database being polled was started with the `-indexrangesize` and/or the `-tablerangesize` parameter. Note that both parameters have a default value of 50. If your database has 60 indexes but the `-indexrangesize` parameter is set for the default, the **Index Details** view will only be able to display data for the first 50 indexes. When the **Index Details** view tries to retrieve data from the `_IndexStat` VST and does not find any, it will return a value of double question marks (??). See *OpenEdge Deployment: Startup Command and Parameter Reference* for more information about the `-indexrangesize` and `-tablerangesize` parameters.

The application area data presented in the above view come from the `_AreaStatus`, `_FileList`, `_ActIOFile`, `_TableStat`, and `_IndexStat` VSTs.

Before Imaging area

The **Before Imaging Area** section contains information about the primary recovery area. Click **Primary Recovery Area** for a more detailed view, as seen in [Figure 3–14](#).

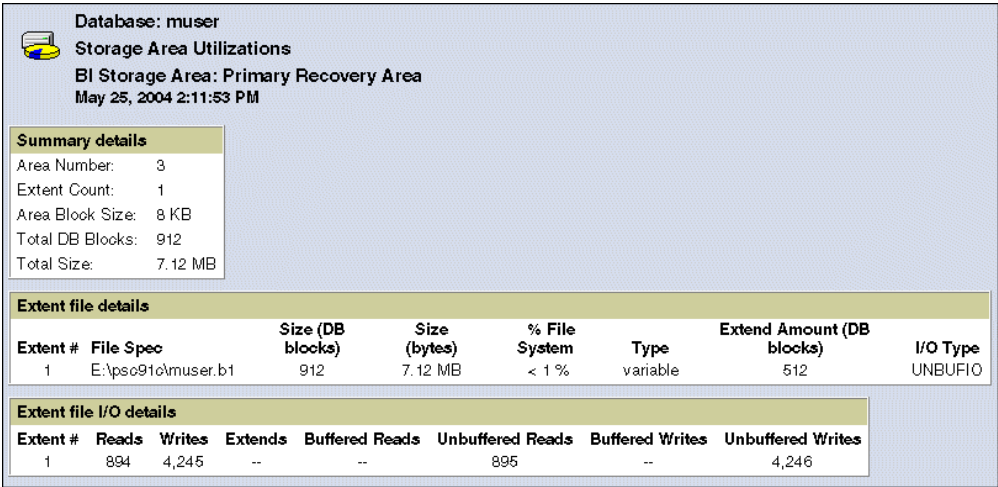



Figure 3–14: Before Image Primary Recovery Area view

The data displayed in the **BI Storage Area** view come from the `_AreaStatus`, `_FileList`, and `_ActIOFile` VSTs.

After Imaging areas

The **After Imaging Areas** section contains information about any defined after image areas. [Figure 3–15](#) shows a sample **AI Storage Area** view.

Database: muser

 Storage Area Utilizations

AI Storage Area: After Image Area 1

May 25, 2004 2:12:14 PM

Summary details

Area Number:	9
Extent Count:	1
Area Block Size:	8 KB
Total DB Blocks:	2,640
Total Size:	20.62 MB

Extent file details

Extent #	File Spec	Size (DB blocks)	Size (bytes)	% File System	Type	Extend Amount (DB blocks)	I/O Type
1	I:\muser2\muser.a1	2,704	21.12 MB	1 %	variable	512	UNBUFIO

Extent file I/O details

Extent #	Reads	Writes	Extends	Buffered Reads	Unbuffered Reads	Buffered Writes	Unbuffered Writes
1	1	3,425	41	--	2	--	4,738

Figure 3–15: After Image Storage Area view

Because only one AI area can be active at a time, additional listed AI areas will have some data columns populated by two question marks. See [Figure 3–8](#) for an example of this.

The data displayed in the **AI Storage Area** view come from the `_AreaStatus`, `_FileList`, and `_ActIOFile` VSTs.

User Activity views

The **User Activity** view is divided into separate frames. The default detail sections are **User Summary** and **User I/O Summary**, as shown in [Figure 3–16](#). The data found in the **User Activity** view come from the `_UserIO` and `_UserLock` VSTs.

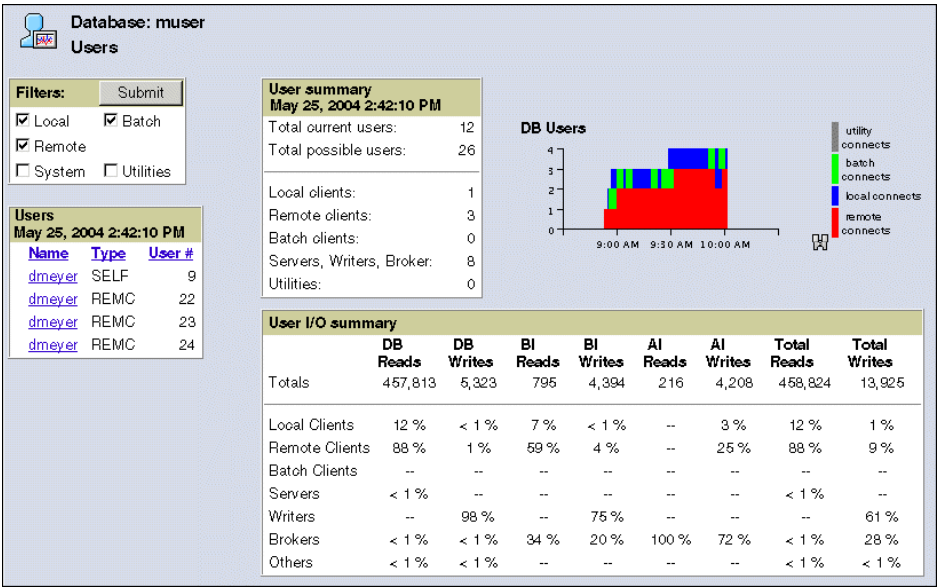


Figure 3–16: User Activity view

Filters and Users sections

The **Filters** section allows you to specify the types of users you want to list. The five types of users are:

- **Local**

Local users are any self-service users that are not batch jobs. Self-service users are denoted in the **Users** section by SELF. A *self-service user* is local to the database being accessed (on the same machine) and is accessing the database directly through shared memory.

- **Remote**

Remote users are those who access the database through a server.

- **System**

System users include the BIW, AIW, APW, and WDog, as well as any brokers and servers. Broker is denoted as BROK and server as SERV in the **Users** section.

- **Batch**

Fathom can only identify self-service batch users. Batch jobs are denoted in the **Users** section by an asterisks (*) after SELF.

- **Utilities**

Utility users include PROMON, PROBKUP, PROSHUT, and Multi-Volume Utility. Utility users are denoted in the **Users** section by MON, BKUP, SHUT, and MVUT, respectively.

Click a user name to display the details section for that user. The details displayed in the details section vary, depending on the user type.

If the user whose name you clicked is a self-service user, the detail section that appears allows you to view transactions and locks for that user, as shown in [Figure 3–17](#).

The screenshot displays a web-based interface for viewing user details. At the top, there are two unchecked checkboxes labeled 'Transactions' and 'Locks', followed by a 'Submit' button. Below this is a section titled 'User details' with a timestamp 'May 25, 2004 2:42:10 PM' and a 'Request Disconnect' button. The user details are listed as follows:

User name:	dmeyer
User #:	9
User type:	SELF
Process ID:	1936
Connect device:	CON:

Figure 3–17: User details

If you select either the **Transactions** check box or the **Locks** check box and then click **Submit**, the additional sections shown in [Figure 3–18](#) appear.

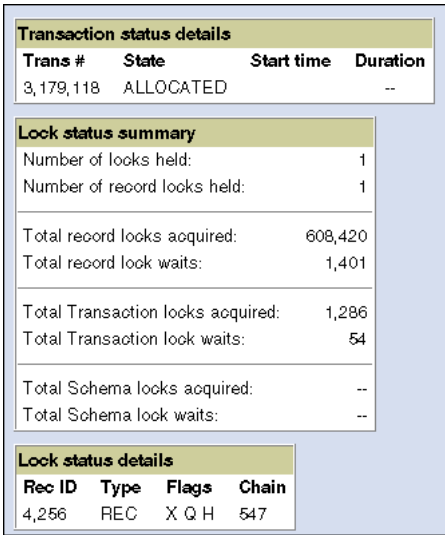


Figure 3–18: Transactions and locks

Note the **Request Disconnect** button shown in [Figure 3–17](#). The **Request Disconnect** button allows you to disconnect users. Before disconnecting the user, Fathom prompts you to confirm.

Record and Index Activity view

The **Record and Index Activity** view presents an overall view of your records and indexes, as shown in [Figure 3–19](#), and also allows you to focus on more specific screens.

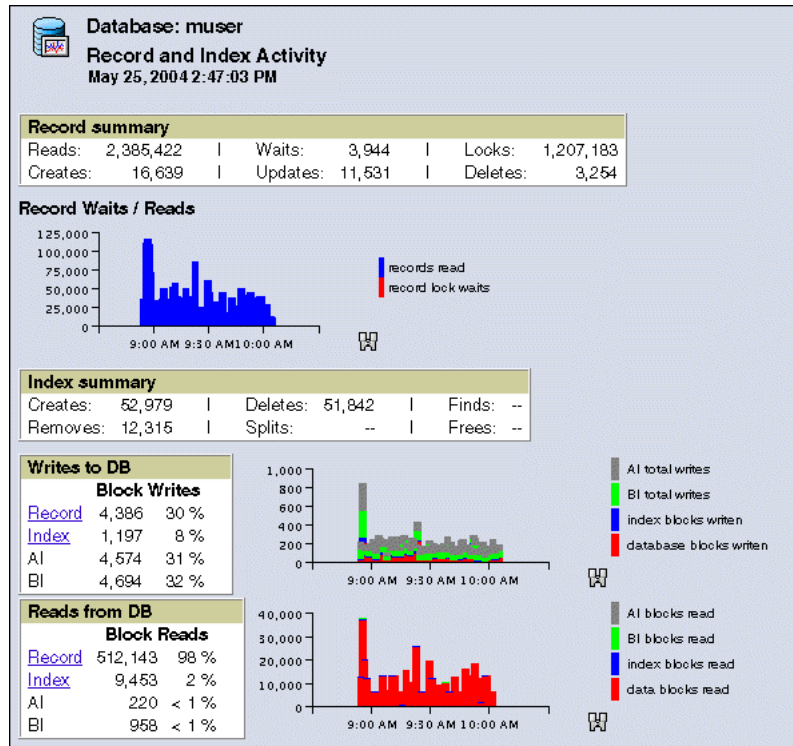


Figure 3–19: Record and Index Activity view

Clicking on **Record** displays the information shown in Figure 3–20.


Database: muser
Record / Table Information
May 25, 2004 2:47:03 PM

All table details

Table #	Name	Area #	Reads	Updates	Creates	Deletes
2	Customer	7	1,232,345	9,964	12,818	3,262
1	Invoice	7	--	--	--	--
3	Item	7	--	--	--	--
8	Local-Default	7	--	--	--	--
4	Order	8	1,167,884	633	3,893	--
5	Order-Line	8	5	--	--	--
9	Ref-Call	7	--	--	--	--
6	Salesrep	7	--	--	--	--
7	State	7	--	--	--	--

Figure 3–20: Record/Table information

Clicking on **Index** displays the information shown in [Figure 3–21](#).

 Database: muser
Index Information
May 25, 2004 2:47:03 PM

All index details									
Index #	Name	Unique	Active	Area #	Reads	Creates	Deletes	Splits	Block Deletes
13	Comments	false	true	7	--	--	--	--	--
14	Country-Post	false	true	7	--	18,026	17,726	--	--
12	Cust-Num	true	true	7	1,786,460	13,056	12,759	3	6
15	Name	false	true	7	--	13,056	12,756	--	--
16	Sales-Rep	false	true	7	--	9,065	8,765	--	--
9	Cust-Num	false	true	7	--	--	--	--	--
10	Invoice-Date	false	true	7	--	--	--	--	--
8	Invoice-Num	true	true	7	--	--	--	--	--
11	Order-Num	false	true	7	--	--	--	--	--
18	Cat-Description	false	true	7	--	--	--	--	--
19	Item-Name	false	true	7	--	--	--	--	--
17	Item-Num	true	true	7	--	--	--	--	--
21	Cust-Order	true	true	8	5,915	3,933	--	--	--
22	Order-Date	false	true	8	--	687	687	--	--
20	Order-Num	true	true	8	1,222,618	--	--	--	--
23	Sales-Rep	false	true	8	--	--	--	--	--
25	item-num	false	true	8	--	--	--	--	--
24	order-line	true	true	8	--	--	--	--	--
26	Sales-Rep	true	true	7	--	--	--	--	--
27	State	true	true	7	--	--	--	--	--
28	default	false	true	6	--	--	--	--	--
29	Call-Num	true	true	7	--	--	--	--	--
30	Cust-Num	true	true	7	--	--	--	--	--
31	Sibling	true	true	7	--	--	--	--	--
32	Txt	false	true	7	--	--	--	--	--

Figure 3–21: Index information

The information displayed in the **Record and Index Activity** view comes from the `_TableStat` and `_IndexStat` VSTs, respectively.

Locks and Latches views

As shown in [Figure 3–22](#) and [Figure 3–23](#), the **Locks and Latches** view is divided into two sections: **Lock summary** and **Latch summary**.

Lock summary				
Looks in use:	411		Most looks in use:	612
Canceled requests:	--		Downgrades:	--
	Requests	Waits	Obtained	Finds
Share Locks:	--	--	--	53,109
Exclusive Locks:	--	--	--	4,390
Record Locks:	--	--	--	
Upgrades:	--	--	--	

Figure 3–22: Lock summary section

Latch summary				
Latch	Type	Holder	Times locked	Time Outs
MTL_MTX	MT_LT_QUEUE	22	399,815	--
MTL_USR	MT_LT_QUEUE	3	2,657	--
MTL_OM	MT_LT_QUEUE	9	2,927,773	33
MTL_BIB	MT_LT_QUEUE	5	534,300	1
MTL_SCC	MT_LT_QUEUE	22	201,751	--
MTL_LKP	MT_LT_QUEUE	9	33,445,232	5
MTL_GST	MT_LT_QUEUE	24	896	--
MTL_TXT	MT_LT_QUEUE	9	529,556	1
MTL_LHT	MT_LT_QUEUE	9	88,042,857	578
MTL_SEQ	MT_LT_QUEUE	--	10	--
MTL_AIB	MT_LT_QUEUE	--	261,638	--
MTL_TXQ	MT_LT_QUEUE	22	332,222	--
MTL_BIW	MT_LT_QUEUE	--	--	--
MTL_LKF	MT_LT_QUEUE	9	4,927,241	--
MTL_BFP	MT_LT_QUEUE	7	1	--
MTL_BHT	MT_LT_QUEUE	7	6,514,693	69
MTL_PWQ	MT_LT_QUEUE	7	14,186	--
MTL_AIW	MT_LT_QUEUE	--	--	--
MTL_CPQ	MT_LT_QUEUE	7	57,266	--
MTL_LRU	MT_LT_QUEUE	9	5,984,520	43
MTL_LR2	MT_LT_QUEUE	--	--	--
MTL_LR3	MT_LT_QUEUE	--	--	--
MTL_LR4	MT_LT_QUEUE	--	--	--
MTL_BF1	MT_LT_QUEUE	9	2,829,744	7
MTL_BF2	MT_LT_QUEUE	9	2,994,875	2
MTL_BF3	MT_LT_QUEUE	7	2,639,155	14
MTL_BF4	MT_LT_QUEUE	9	4,187,395	14
MTL_BF5	MT_LT_QUEUE	--	--	--
MTL_BF6	MT_LT_QUEUE	--	--	--
MTL_BF7	MT_LT_QUEUE	--	--	--

Figure 3–23: Latch summary section

The information found in the **Lock** and **Latch summary** views comes from the `_ActLock` and `_Latch` VSTs.

Transactions view

Figure 3–24 shows the **Transactions** view. All the data associated with transactions currently being processed within the database display in this single view. The information for this view comes from the `_Trans VST` table. This view presents a snapshot of this data at the date and time specified at the top of the page. To avoid viewing stale data on this page, select the **Refresh** icon periodically.

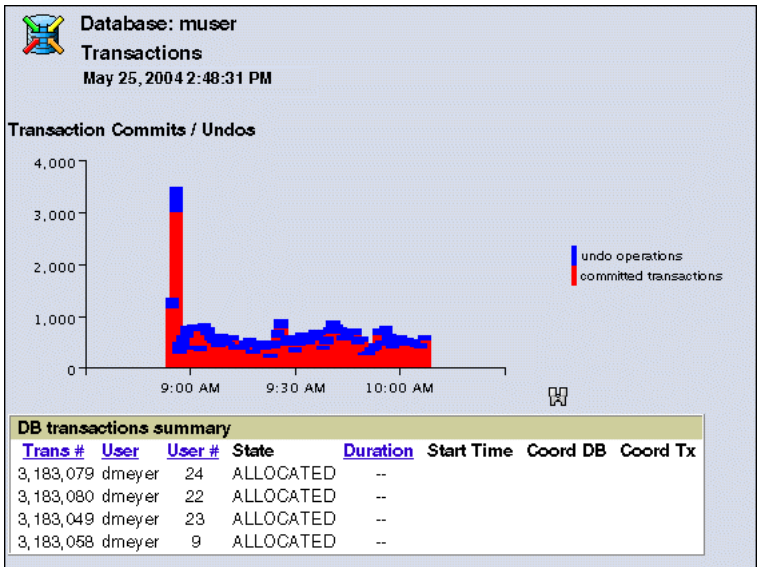


Figure 3–24: Transactions page

The display includes the following columns:

- **Trans** — The transaction number.
- **User** — The user who entered the transaction. This information reflects the information available in the **User Activity** view.
- **User #** — The number of the user listed in the **User** column. This information reflects the information available in the **User Activity** view.
- **State** — The state of the transaction (either **Allocated**, **Active**, or **Committing**). Allocated means the transactions have been “announced” to the database, but not yet started.

The **Active** state is read to determine the data for the **Duration** and **Start Time**.

- **Duration** — How much time has passed since the transaction started. Displayed for active transactions.
- **Start Time** — When the transaction started. Displayed for active transactions.
- **Coord DB** — The name of the coordinator database. Displays if two-phase commit is in use.
- **Coord Tx** — The number of the transaction on the coordinator database. Displays if two-phase commit is in use.

Memory Resources view

The **Memory Resources** view is divided into the following sections: **Buffers summary**, **Shared memory summary**, **Space management summary**, and **Resources summary**. See [Figure 3–25](#), [Figure 3–26](#), [Figure 3–27](#), and [Figure 3–28](#) for sample views of each section.

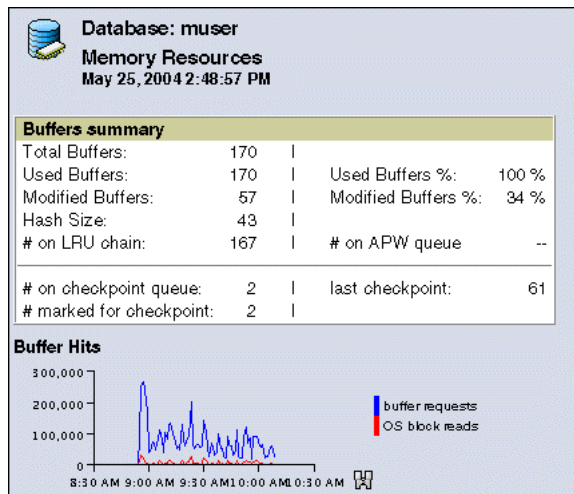


Figure 3–25: Buffers summary section

Shared memory summary				
#	OS ID	Segment Size	Bytes Used	Bytes Free
1	10,092,544	1.31 MB	1.24 MB	73 KB

Figure 3–26: Shared memory summary section

Space management summary				
DB Extends:	16	I	Record Bytes:	7.29 MB
Taken from Free:	32	I	Returned to Free:	6
RM's examined:	19,147	I	RM's removed:	842
Allocs from RM:	17,760	I	Allocs from Free:	32
Add RM to Front:	881	I	Add RM to Back:	--
RM Front to Back:	545	I	RM Blocks Locked:	16,199

Figure 3–27: Space management summary section

Resources summary		
Resource	Locks	Waits
Shared Memory	--	--
Record Lock	1,248,185	4,108
Schema Lock	--	--
Trans Commit	13,041	55
DB Buf I Lock	125,501	--
Record Get	1,344,954	--
DB Buf Read	535,527	--
DB Buf Write	5,877	--
DB Buf Backup	--	--
DB Buf S Lock	5,820,943	44
DB Buf X Lock	67,164	14
DB Buf Avail	--	--
12	--	--
BI Buf Read	988	--
BI Buf Write	4,915	193
AI Buf Read	220	--
AI Buf Write	4,792	--
17	--	--
TXE Share Lock	130,059	--
TXE Update Lock	32,134	--
TXE Commit Lock	7,793	7
TXE Excl Lock	--	--
22	--	--

Figure 3–28: Resources summary section

Data populating the **Database Resources** views come from the `_ActBuffer`, `_BuffStatus`, `_Segments`, `_ActSpace`, and `_Resrc` VSTs.

Page Writers view

The **Page Writers** view displays data about your Before-image, After-image, and Asynchronous Page Writers, as well as checkpoint information. The **Page Writers** view is divided into the following sections: **BI summary**, **AI summary**, **Asynchronous Page Writer summary**, and **Checkpoint summary**. See [Figure 3–29](#), [Figure 3–30](#), [Figure 3–31](#), and [Figure 3–32](#) for samples of each section.

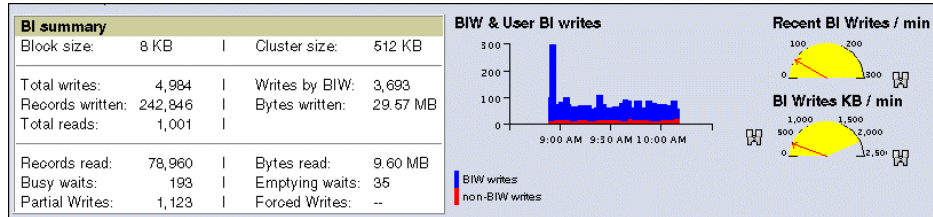


Figure 3–29: Before Image summary section

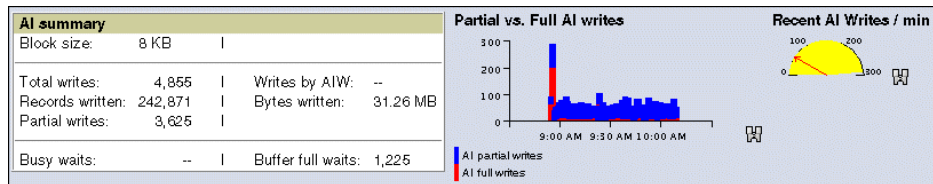


Figure 3–30: After Image summary section

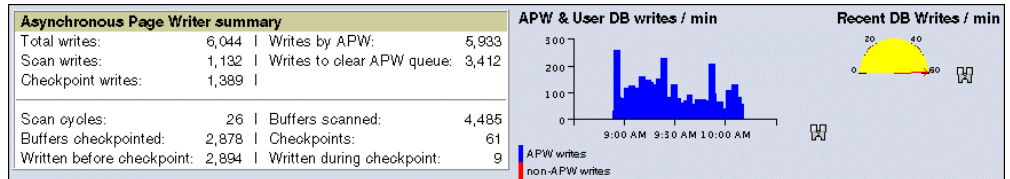


Figure 3–31: Asynchronous Page Writer summary section

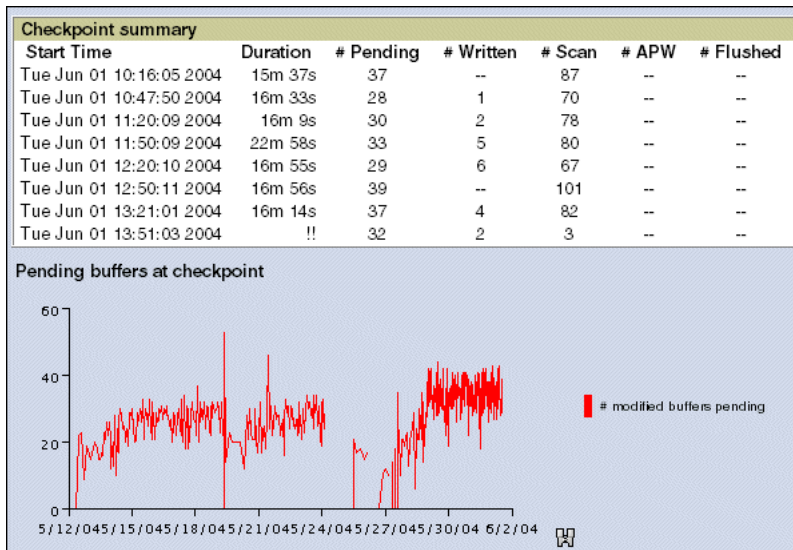


Figure 3–32: Checkpoint summary section

The **Checkpoint summary** page contains the following columns:

- **Start Time** — The time the checkpoint started.
- **Duration** — How long the checkpoint lasted.
- **# Pending** — Number of database buffers not yet written to disk.
- **# Written** — Number of buffers written to disk.
- **# Scan** — Number of buffers written during the scan.
- **# APW** — Number of buffers written from the APW queue.
- **# Flushed** — Number of buffers flushed at the checkpoint.

The data displayed in the **Writer** sections comes from the following VSTs:

- `_ActBILog` (BI Summary View)
- `_ActAILog` (AI Summary View)
- `_ActPW` (APW Summary View)
- `_Checkpoint` (Checkpoint Summary View)

Start Parameters view

Figure 3–33 shows the database **Start Parameters** view. This single-view page displays a subset of the parameters used to start the database. The values that display are static; they will change only if the database is stopped or started and you explicitly change a setting.


 Database: muser Start Parameters May 25, 2004 3:01:54 PM		
DB start parameters		
	PSC Option Setting	Meaning
Max Clients (-Ma):	-Ma 5	
Max Servers (-Mn):	-Mn 4	
Max Users (-n):	-n 21	
Buffers (-B):	-B 168	
Direct I/O (-directio):		No
Crash Protection (-i):		Full
Lock Table (-L):	-L 8192	
Spin Locks (-spin):	-spin 0	
BI Buffers (-bibufs):	-bibufs 5	
BI commit delay (-Mf):	-Mf 3	
BI I/O (-r):		Normal
BI aging before reuse (-G):	-G 60	
BI block size (-biblocksize):	-biblocksize 8192	8 KB
BI cluster size (-bi):	-bi 512	512 KB
AI Buffers (-aibufs):	-aibufs 5	
AI block size (-aiblocksize):	-aiblocksize 8192	8 KB
APW Buffers (-pwscan):	-pwscan 1	
APW queue check time (-pwqdelay):	-pwqdelay 100	(in milliseconds)
APW scan (-pwsdelay):	-pwsdelay 1	
APW max writes (-pwwmax):	-pwwmax 25	

Figure 3–33: Database Start Parameters page

The parameters that appear in the **DB start parameters** column originate from the database `_StartUp VST` table. The second column, **PSC Option Setting**, shows the values entered in the command line set for each option or system-defined default values taken by the program at the starting of the database. Blanks indicate that the parameter is not used.

The **Meaning** column shows additional interpretive information, where appropriate, that is intended to provide brief text commentary about the start parameters that are set for the database.

General Details view

The **General Details** view provides summary information about your database. [Figure 3–34](#) shows a sample **General Details** page.

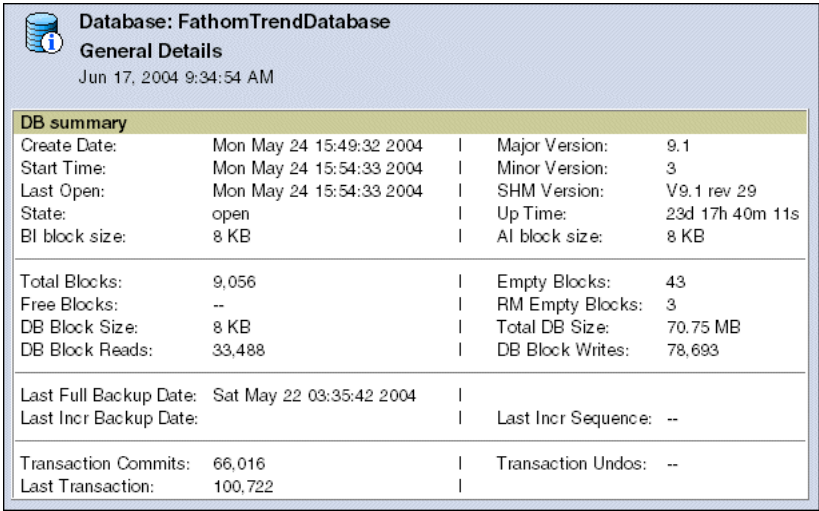


Figure 3–34: General Details view

Advanced views

There are two advanced views: **Raw VST Data** and **Raw System Table Data**.

Note: Advanced views are provided for users who have worked with and understand raw data.

[Figure 3–35](#) shows an example of the **Raw VST Data** page.

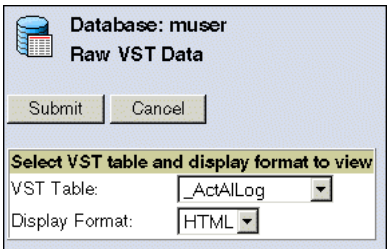
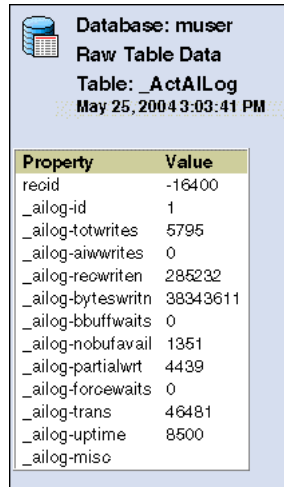


Figure 3–35: Raw VST Data page

Submittal of the page shown in [Figure 3–35](#) returns the data shown in [Figure 3–36](#).

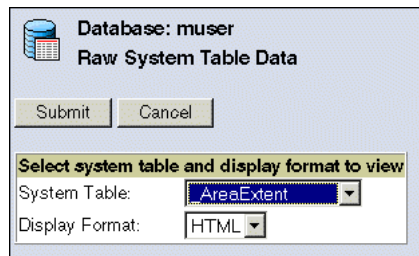


Database: muser
Raw Table Data
Table: _ActAllLog
 May 25, 2004 3:03:41 PM

Property	Value
recid	-16400
_ailog-id	1
_ailog-totwrites	5795
_ailog-aiwwrites	0
_ailog-reowriten	285232
_ailog-byteswritn	38343611
_ailog-bbuffwaits	0
_ailog-nobufavail	1351
_ailog-partialwrt	4439
_ailog-forcewaits	0
_ailog-trans	46481
_ailog-uptime	8500
_ailog-misc	

Figure 3–36: _ActAllLog Data page

[Figure 3–37](#) shows a sample **Raw System Table Data** page.



Database: muser
Raw System Table Data

Submit Cancel


Select system table and display format to view

System Table:

Display Format:

Figure 3–37: Raw System Table Data page

Submittal of the page shown in [Figure 3–37](#) returns the page shown in [Figure 3–38](#).



Database: muser

Raw Table Data

Table: _AreaExtent

May 25, 2004 3:03:55 PM

recid	_area-recid	_extent-version	_extent-number	_area-number	_extent-type	_extent-attrib	_extent-system	_extent-size	_extent-path	_extent-misc
-72	128	1	1	1	6	0	0	0	E:\pso910\muser.db	?,?,?,?,?,?,?
-72	130	1	1	3	6	0	0	0	E:\pso910\muser.b1	?,?,?,?,?,?,?
-72	132	1	1	6	5	0	0	0	E:\pso910\muser.d1	?,?,?,?,?,?,?
-72	134	1	1	7	5	0	0	0	E:\pso910\muser_7.d1	?,?,?,?,?,?,?
-72	136	1	1	8	37	0	0	6976	m:\muser3 \muser_8.d1	?,?,?,?,?,?,?
-72	136	1	2	8	37	0	0	40960	m:\muser3 \muser_8.d2	?,?,?,?,?,?,?
-72	136	1	3	8	5	0	0	0	m:\muser3 \muser_8.d3	?,?,?,?,?,?,?
-72	138	1	1	9	4	0	0	0	l:\muser2\muser.a1	?,?,?,?,?,?,?
-72	140	1	1	10	4	0	0	0	l:\muser2\muser.a2	?,?,?,?,?,?,?
-72	142	1	1	11	4	0	0	0	l:\muser2\muser.a3	?,?,?,?,?,?,?

Figure 3–38: Sample Raw Table Data page

Database Maintenance Job Templates

Fathom Management supplies several database maintenance job templates to help you set up and schedule routine maintenance activities for your OpenEdge database. This chapter identifies these templates and describes how to set up and run each of them, as outlined in the following sections:

- [Database maintenance job templates](#)
- [Creating database maintenance job instances](#)
- [Advanced information](#)

Note: This chapter assumes that you are familiar with the job information presented in the *Resource Monitoring Guide*.

Database maintenance job templates

Database maintenance job templates are Fathom-supplied, specialized job templates from which you can create job instances to address fundamental OpenEdge database maintenance activities. Fathom Management provides these predefined job templates to:

- Ensure that you can quickly and easily set up and initiate maintenance routines for your OpenEdge databases.
- Standardize the information you want used when these routine activities are performed.
- Define schedules for the various types of maintenance routines you perform for your OpenEdge databases.

Using these job templates allows you to institute what can be considered a Fathom “best practices” approach to ensuring specific database activities are accomplished routinely for your OpenEdge database.

Note: Fathom Management also provides a database maintenance job template to be used only with the FathomTrendDatabase. This job template is called **Trend Data Compaction** (dataCompaction). See the [FathomTrendDatabase Guide and Reference](#) for information about this job.

Displaying database job templates

One way to view the database maintenance job templates is to select **Jobs** from the menu bar. In the list frame, expand the **Job Templates** category, as shown in [Figure 4–1](#).

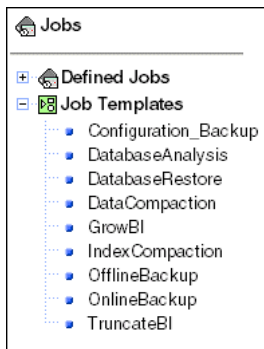


Figure 4–1: Database maintenance job templates

The following items display under the **Job Templates** category:

- Predefined maintenance job templates.
- The FathomTrendDatabase **Trend Data Compaction** template (displayed as **DataCompaction** in the list frame).
- Any user-defined job templates.

To display the predefined job template pages associated with each of these database management activities, select a template from those listed. A summary of the current template values displays in the detail frame. You can review or edit the values defined for the database maintenance templates as you would any user-created job template.

Note: Template jobs can only be run against managed databases. In order to run a job against a scripted database, you must create your own job, without a template. See the [Resource Monitoring Guide](#) for more information on creating jobs without using templates.

General considerations

You work with database maintenance jobs just as you do any jobs you create yourself. For example, once an instance of a database maintenance job is created, you can run it immediately or schedule it to run at a later date.

Keep the following points in mind when working with job instances:

- By default, the database management jobs are enabled to run as job actions. This option supports job chaining. See the [Resource Monitoring Guide](#) for a detailed explanation of job chaining.
- Database job instances use environment variables common to all jobs, and some that are defined specifically for the type of database maintenance activity you are performing. For information about common environmental variables, see the [Resource Monitoring Guide](#). See the “[Environment variables](#)” section on page 4–19 for details about environment variables specific to each database maintenance job.
- All predefined default field values are editable, including the template menu categories.
- Note that the name of your job cannot contain spaces or the following special characters: %, &, \$, #.
- Required fields on the job’s property page appear in red.

For job instances derived from a database maintenance job template, you can:

- Access the database maintenance job template either directly from the **Job Details Home** page or from the **Job Templates** category in the list frame. Keep in mind that you can only review and edit the template if you access it from the list frame; you cannot set up a job instance of a given template from the list frame.
- Set up the **Job Completion Actions and Alerts** page.
- Review specific job values on the **Job Summary** page. This page automatically displays once you complete either the **Job Properties** page or the **Job Completion Actions and Alerts** page.

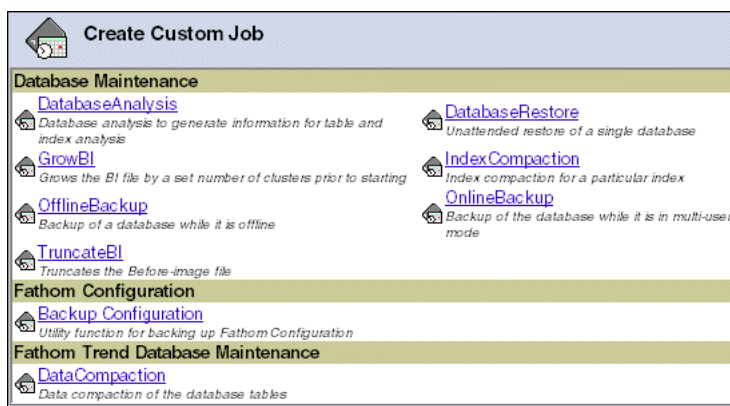
Creating database maintenance job instances

To create a job instance from one of the database maintenance templates, you must access the specific template details from the **Jobs** page.



To access job templates for setting up a job instance:

1. From the menu bar, select **Jobs**. The **Fathom Jobs** page appears.
2. Select **Create Job from a Template**. The **Create Custom Job** page appears:



The **Database Maintenance**, **Fathom Configuration**, and **Fathom Trend Database Maintenance** categories include all the Fathom-supplied jobs. You can create additional templates to add to these categories or create additional categories. For details about creating user-defined templates, see the *Resource Monitoring Guide*.

3. Select the database maintenance job from which you want to create a job instance. The **Job Properties** page appears.

Note: When the **Job Properties** page appears, you might find that several fields already display values. This is one of the main benefits of the template; you only need to supply the data unique to this job instance.

4. Complete the **Job Properties** page. Use the following information to find details about each database maintenance job template's **Properties** page:

For details about the. . .	See the. . .
OfflineBackup template	“Database Backup jobs” section on page 4–5.
OnlineBackup template	“Database Backup jobs” section on page 4–5.
DatabaseRestore template	“Database Restore job” section on page 4–8.
TruncateBI template	“Truncate BI job” section on page 4–10.
GrowBI template	“Grow BI job” section on page 4–12.
DatabaseAnalysis template	“Database Analysis job” section on page 4–14.
IndexCompaction template	“Index Compaction job” section on page 4–15.
Backup Configuration template	“Backup Configuration job” section on page 4–16.
DataCompaction template	<i>FathomTrendDatabase Guide and Reference.</i>

Database Backup jobs

The job page and options to set up and run an online database backup are identical to an offline backup.



To complete the Database Backup Properties page:

1. Provide values for the **Name** and **Description**.
2. From the list of available databases, select the database for which you want to perform this database backup.

3. Identify the backup device you are using:
 - If you select **Tape**, you must also define the absolute path for the tape device.
 - If you select **Disk**, you must also specify the directory and the filename. If you want to overwrite an existing backup at this location, select the **Overwrite file** option.
4. In the **Volume size (-vs)** field, identify the size, in database blocks, that you want the backup to be before prompting for another volume.

If you select the **Volume size (-vs)** parameter, the following happens:

- The Backup job looks for a file in your work directory with the name of <resource name>-backup.txt. This file should contain names for each backup file. If this file exists, the Backup job uses it as input to the OpenEdge Backup Utility (PROBKUP).

The following figure shows an example <resource name>-backup.txt file:

```
e:\dbwork/ftd.bak1
e:\dbwork/ftd.bak2
e:\dbwork/ftd.bak3
e:\dbwork/ftd.bak4
e:\dbwork/ftd.bak5
e:\dbwork/ftd.bak6
e:\dbwork/ftd.bak7
e:\dbwork/ftd.bak8
e:\dbwork/ftd.bak9
e:\dbwork/ftd.bak10
e:\dbwork/ftd.bak11
e:\dbwork/ftd.bak12
e:\dbwork/ftd.bak13
e:\dbwork/ftd.bak14
e:\dbwork/ftd.bak15
e:\dbwork/ftd.bak16
e:\dbwork/ftd.bak17
e:\dbwork/ftd.bak18
e:\dbwork/ftd.bak19
e:\dbwork/ftd.bak20
e:\dbwork/ftd.bak21
```

- If this file does not exist, the Backup job creates the file in the working directory. The file is the name of the backup (entered in [Step 1](#)) with a number from 1 to 99 appended to it. The Backup job uses this file as input to PROBKUP.
- Any files created by the Backup job remain after the program ends. The Backup job does not delete them.

5. Complete the following fields:

- **Blocking factor (-bf)** — Specify the blocking factor size used to flush buffers out of the backup device.
- **Redundancy factor (-red)** — Specify the redundancy count to create redundancy in the backup.
- **Incremental overlap (-io)** — Identify how many previous backups you want included in this incremental backup. Use this field simultaneously with the **Incremental** check box.
- **Incremental** — Check this box to back up the blocks that have changed since the last backup. Use this field simultaneously with the **Incremental overlap (-io)** parameter field.
- **Estimate size** — Check this box to generate a report that estimates the size of the backup.

Note: This setting *does not* back up the database.

- **Verbose listing** — Check this box to have Fathom provide output every 10 seconds concerning the status of the backup.
 - **Compression** — Check this box to indicate whether you want to compress data blocks as they are backed up.
 - **No recovery** — Check this box if you do not want the database to be able to roll forward an AI file. This option is used for mirrored backups.
6. Verify the name of the command's current working directory in the **Working Directory** field. If specified, this directory must exist. This property defaults to the working directory defined at installation.
7. To append subsequent messages to the `.out` or `.err` files, select the **Append** option. The **Output file** field indicates where messages will be retained.
8. Select the **Debug log file** option to obtain diagnostic details that help debug job properties when setting up a job. For example, you can set this option and use the **Run Now** feature on the **Job Summary** page. Once you have submitted the job, Fathom makes debug data available through the debug log file link on the **Job Summary** page.

9. If you intend to set up actions and alerts for this job, click the **Edit** button associated with the **Completion Actions and Alerts** field to display the **Job Completion Actions and Alerts** page. The **Job Completion Actions and Alerts** page appears. See the [Resource Monitoring Guide](#) for more information about the **Job Completion Actions and Alerts** page.
10. Click **Save**. The **Job Summary** page appears. You can now schedule the job or run it immediately. You can also edit advanced information. See the [“Advanced information”](#) section on page 4–18 for steps on editing advanced information.

Database Restore job

The Database Restore job restores an OpenEdge backup to the location specified in the existing structure for that database. If the file structure does not exist, then the restore is performed to the existing directory for all of the areas and extents.



To complete the Database Restore Properties page:

1. Provide values for the **Name** and **Description** fields.
2. Identify the restore device you are using:
 - If you select **Tape**, you must also define the absolute path for the tape device.
 - If you select **Disk**, you must also specify the directory and file name.

Note: If you used the volume size (-vs) parameter when backing up your database to disk, you must specify the file that contains the names of each volume. The Restore job will pass this file to the OpenEdge Restore (PROUTIL) utility. The name of the file should be <database name>-restore.txt. The contents of the restore file should be the same as the contents of the backup file passed to PROBKUP. See [Step 4](#) under the [“Database Backup jobs”](#) section on page 4–5 for more information about the backup file.

3. Provide the name of the database you intend to restore in the **Restore to** field. This name is the full path and name of the database to which you want to restore. If you want to overwrite an existing database at this location, select the **Overwrite database** option.

4. Complete the following fields:

- **Partial verify (-vp)** — Select if you want to ensure that a partial verification of the completed backup occurs. It only ensures that the backup can be read.

Note: This option *does not* restore the database.

- **Full verify (-vf)** — Select if you want a full verification of the completed backup. This process ensures that a block-by-block comparison is performed between the backup and the database.

Note: This option *does not* restore the database.

- **List structure (-list)** — Select to create a structure file from the backup that was completed.

Note: This option *does not* restore the database.

- **No verification** — Deselect if you want verification. Note that **No verification**, indicating that no verification process will occur, is the default selection.

5. Verify the name of the command's current working directory in the **Working Directory** field. If specified, this directory must exist. This property defaults to the working directory defined at installation.
6. To append subsequent messages to the `.out` or `.err` files, select the **Append** option. The **Output file** field indicates where messages will be retained.
7. Select the **Debug log file** option to obtain diagnostic details that help debug job properties when setting up a job. For example, you can set this option and use the **Run Now** feature on the **Job Summary** page. Once you have submitted the job, Fathom makes debug data available through the debug log file link on the **Job Summary** page.

8. If you intend to set up actions and alerts for this job, click the **Edit** button associated with the **Completion Actions and Alerts** field to display the optional **Job Completion Actions and Alerts** page. The **Job Completion Actions and Alerts** page appears. See the *Resource Monitoring Guide* for more information about the **Job Completion Actions and Alerts** page.
9. Click **Save**. The **Job Summary** page appears. You can now schedule the job or run it immediately. You can also edit advanced information. See the “[Advanced information](#)” section on page 4–18 for steps on editing advanced information.

Truncate BI job

The Truncate BI (before image) job allows you to truncate the BI file, shrinking it to zero bytes. This job can be used in processing, if needed; however, truncation is typically bundled with Grow BI for performance reasons.



To complete the Truncate BI Properties page:

1. Provide values for the **Name** and **Description** fields.
2. From the **Resources** list, select the database whose BI file you want to truncate.
3. Identify the operating system user account in the **User name** or **Group** field. It is not necessarily the same as your Fathom user account name. If specified, it must be a valid account on the server machine (or server domain) where the AdminServer and Fathom are running. On Windows the name can also include a domain.
4. Enter the password of the specified user in the **Password** field. If the **User name** or **Group** does not have an associated password, this field must be left blank, or an error message is generated. Provide a valid user name and password in the account information field.
5. Complete the **Job specification** section as follows:
 - a. Enter the command to be executed in the **Command** field. This can be any user-defined command that would typically execute from an OS shell. The name can include a full or relative pathname. You can use environment variables such as %DLC% or \$DLC. This value will already be defined if you are creating a job instance from a template.

- b. Define the input parameter to the command in the **Command parameters** field. The list of parameters takes the same format as from an OS shell. Additionally, you can use environment variables (\$SHELL or %WINDIR%, for example) and Windows registry values (@SOFTWARE\PSC\FATHOM\3.0A\10.0B\FATHOMINIT, for example).
- c. Verify the name of the command's current working directory in the **Working directory** field. If specified, this directory must exist. This property defaults to the working directory defined at installation.
- d. Enter the name of the **Input file** to be used for read redirection with a job's command. This is typically used for any keyboard input the command might require. This field is optional.
- e. To append subsequent messages to the .out or .err files, select the **Append** option. The **Output file** fields indicate where messages will be retained.

Note: When creating or editing a job, Fathom supplies the .out and .err filenames for you. The filenames will be the same as the job name with a different suffix. You can change the filename, or remove it if you do not want to create the output files. Also, if you edit the job name, Fathom will not change the filenames to reflect the new job name. You should review these filenames *before* saving the page to ensure the job creates the expected output files.

- 6. In the **Environment name=value pairs** field, define environment variables to be set in the process context of the task that runs to execute the specified job. These values are application-specific, user-definable variables. For a list of environment variables available for the job, run the job with the **Debug log file** option on. The debug log file lists all environment variables and, if applicable, their values. Environment variables are preceded by **env** in the log file.
- 7. Select the **Debug log file** option to obtain diagnostic details that help debug job properties when setting up a job. For example, you can set this option and use the **Run Now** feature on the **Job Summary** page. Once you have submitted the job, Fathom makes debug data available through the debug log file link on the **Job Summary** page.
- 8. Select the **Indicate if the job can be used as an action** check box if you want this job to appear in the list of actions that can be selected for execution from the **Alerts and Job completion action** page.

9. If you intend to set up actions and alerts for this job, click the **Edit** button associated with the **Completion Actions and Alerts** field to display the **Job Completion Actions and Alerts** page. The **Job Completion Actions and Alerts** page appears. See the [Resource Monitoring Guide](#) for more information about the **Job Completion Actions and Alerts** page.
10. Click **Save**. The **Job Summary** page appears. You can now schedule the job or run it immediately.

Grow BI job

Grow BI job instances should be used if the BI file is truncated. Grow BI jobs preformat the BI file and result in better performance.



To complete the Grow BI Properties page:

1. Provide values for the **Name** and **Description** fields.
2. From the **Resources** list, select the database whose BI file you want to grow.
3. Identify the operating system user account in the **User name** or **Group** field. It is not necessarily the same as your Fathom user account name. If specified, it must be a valid account on the server machine (or server domain) where the AdminServer and Fathom are running. On Windows the name can also include a domain.
4. Enter the password of the specified user in the **Password** field. If the **User name** or **Group** does not have an associated password, this field must be left blank, or an error message is generated. Provide a valid user name and password in the account information field.
5. Complete the **Job specification** section as follows:
 - a. Enter the command to be executed in the **Command** field. This can be any user-defined command that would typically execute from an OS shell. The name can include a full or relative pathname. You can use environment variables such as %DLC% or \$DLC. This value will already be defined if you are creating a job instance from a template.
 - b. Define the input parameter to the command in the **Command parameters** field. The list of parameters takes the same format as from an OS shell. Additionally, you can use environment variables (\$SHELL or %WINDIR%, for example) and Windows registry values (@SOFTWARE\PSC\FATHOM\3.0A\10.0B\FATHOMINIT, for example).

- c. Verify the name of the command's current working directory in the **Working directory** field. If specified, this directory must exist. This property defaults to the working directory defined at installation.
- d. Enter the name of the **Input file** to be used for read redirection with a job's command. This is typically used for any keyboard input the command might require (optional).
- e. To append subsequent messages to the `.out` or `.err` files, select the **Append** option. The **Output file** fields indicate where messages will be retained.

Note: When creating or editing a job, Fathom supplies the `.out` and `.err` filenames for you. The filenames will be the same as the job name with a different suffix. You can change the filename, or remove it if you do not want to create the output files. Also, if you edit the job name, Fathom will not change the filenames to reflect the new job name. You should review these filenames *before* saving the page to ensure the job creates the expected output files.

- 6. In the **Environment name=value pairs** field, define environment variables to be set in the process context of the task that runs to execute the specified job. These values are application-specific, user-definable variables. For a list of environment variables available for the job, run the job with the **Debug log file** option on. The debug log file lists all environment variables and, if applicable, their values. Environment variables are preceded by **env** in the log file. For environmental variables specific to the Grow BI job, see [Table 4-1](#).
- 7. Select the **Debug log file** option to obtain diagnostic details that help debug job properties when setting up a job. You can set this option and use the **Run Now** feature on the **Job Summary** page. Once you have submitted the job, Fathom makes debug data available through the debug log file link on the **Job Summary** page.
- 8. Select the **Indicate if the job can be used as an action** check box if you want this job to appear in the list of actions that can be selected for execution from the **Alerts and Job completion action** page.

9. If you intend to set up actions and alerts for this job, click the **Edit** button associated with the **Completion Actions and Alerts** field to display the **Job Completion Actions and Alerts** page. The **Job Completion Actions and Alerts** page appears. See the [Resource Monitoring Guide](#) for more information about the Job Completion Actions and Alerts page.
10. Click **Save**. The **Job Summary** page appears. You can now schedule the job or run it immediately.

Database Analysis job

The Database Analysis job runs against an OpenEdge database and loads the information into the FathomTrendDatabase. This information can be used for various purposes, including index compaction and data analysis.



To complete the Database Analysis Properties page:

1. Provide values for the **Name** and **Description** fields.
2. Select the database for which you want to perform this analysis from the **Resources** list.
3. Select the **Enable rule validation** check box to indicate that you want this job to test your rules. **Utilization %** and **Block number**, two subfields of the **Enable rule validation** option, are active only if you select this option. Complete these two fields as follows:
 - a. In the **Utilization %** field, specify the percentage of utilization of the blocks for the index to validate against. If the index utilization percentage goes below this number, you could set up an alert to trigger.
 - b. In the **Block number** field, specify the threshold for the number of blocks that the index must contain for the rule to be run.
4. Select the **Run index compaction** check box if you want index compaction run for the database during this job. The **Compaction %** field specifies the percentage to which indexes will be packed when index compaction is run.
5. Verify the command's current working directory in the **Working directory** field. If specified, this directory must exist. This property defaults to the working directory defined at installation.
6. To append subsequent messages to the `.out` or `.err` files, select the **Append** option. The **Output file** field indicates where messages will be retained.

7. Select the **Debug log file** option to obtain diagnostic details that help debug job properties when setting up a job. For example, you can set this option and use the **Run Now** feature on the **Job Summary** page. Once you have submitted the job, Fathom makes debug data available through the debug log file link on the **Job Summary** page.
8. Select the **Indicate if the job can be used as an action** check box if you want this job to appear in the list of actions that can be selected to be executed from the **Alerts and Job completion action** page.
9. If you intend to set up actions and alerts for this job, click the **Edit** button associated with the **Completion Actions and Alerts** field to display the **Job Completion Actions and Alerts** page. The **Job Completion Actions and Alerts** page appears. See the [Resource Monitoring Guide](#) for more information about the **Job Completion Actions and Alerts** page.
10. Click **Save**. The **Job Summary** page appears. You can now schedule the job or run it immediately. You can also edit advanced information. See the “[Advanced information](#)” section on page 4–18 for steps on editing advanced information.

Index Compaction job

The Index Compaction job compacts indexes to optimal compression, and can be run online or offline. To determine which indexes to compact, run database analysis against the database whose indexes you intend to compact.



To complete the Index Compaction Properties page:

1. Provide values for the **Name** and **Description** fields.
2. Select the database for which you are performing this activity.
3. Complete the following fields:
 - **Table name** — Identify the table on which you intent to perform the index compaction activity.
 - **Index name** — Identify the index name for which you intend to perform the index compaction activity.
 - **Compaction %** — Specify the percentage to which indexes will be packed when index compaction is run.

4. Verify that the **Working directory** field identifies the command's current working directory. If specified, this directory must exist. This property defaults to the working directory defined at installation.
5. To append subsequent messages to the `.out` or `.err` files, select the **Append** option. The **Output file** field indicates where messages will be retained.
6. Select the **Debug log file** option to obtain diagnostic details that help debug job properties when setting up a job. For example, you can set this option and use the **Run Now** feature on the **Job Summary** page. Once you have submitted the job, Fathom makes debug data available through the debug log file link on the **Job Summary** page.
7. Select the **Indicate if the job can be used as an action** check box if you want this job to appear in the list of actions that can be selected to be executed from the **Alerts and Job completion action** page.
8. If you intend to set up actions and alerts for this job, click the **Edit** button associated with the **Completion Actions and Alerts** field to display the **Job Completion Actions and Alerts** page. The **Job Completion Actions and Alerts** page appears. See the [Resource Monitoring Guide](#) for more information about the **Job Completion Actions and Alerts** page.
9. Click **Save**. The **Job Summary** page appears. You can now schedule the job or run it immediately. You can also edit advanced information. See the [“Advanced information”](#) section on page 4–18 for steps on editing advanced information.

Backup Configuration job

The Backup Configuration job backs up the Fathom configuration for a database.



To complete the Backup Configuration Properties page:

1. Provide values for the **Name** and **Description**.
2. From the **Resources** menu, select the database whose Fathom configuration you want to backup.
3. Identify the operating system user account in the **User name** or **Group** field. It is not necessarily the same as your Fathom user account name. If specified, it must be a valid account on the server machine (or server domain) where the AdminServer and Fathom are running. On Windows the name can also include a domain.

4. Enter the password of the specified user in the **Password** field. If the **User name** or **Group** does not have an associated password, this field must be left blank, or an error message is generated. Provide a valid user name and password in the account information field.
5. Complete the **Job specification** section as follows:
 - a. Enter the command to be executed in the **Command** field. This can be any user-defined command that would typically execute from an OS shell. The name can include a full or relative pathname. You can use environment variables such as %DLC% or \$DLC. This value will already be defined if you are creating a job instance from a template.
 - b. Define the input parameter to the command in the **Command parameters** field. The list of parameters takes the same format as from an OS shell. Additionally, you can use environment variables (\$SHELL or %WINDIR%, for example) and Windows registry values (@SOFTWARE\PSC\FATHOM\3.0A\10.0B\FATHOMINIT, for example).
 - c. Verify the name of the command's current working directory in the **Working directory** field. If specified, this directory must exist. This property defaults to the working directory defined at installation.
 - d. Enter the name of the **Input file** to be used for read redirection with a job's command. This is typically used for any keyboard input the command might require. This field is optional.
 - e. To append subsequent messages to the .out or .err files, select the **Append** option. The **Output file** fields indicate where messages will be retained.

Note: When creating or editing a job, Fathom supplies the .out and .err filenames for you. The filenames will be the same as the job name with a different suffix. You can change the filename, or remove it if you do not want to create the output files. Also, if you edit the job name, Fathom will not change the filenames to reflect the new job name. You should review these filenames *before* saving the page to ensure the job creates the expected output files.

6. In the **Environment name=value pairs** field, define environment variables to be set in the process context of the task that runs to execute the specified job. These values are application-specific, user-definable variables. For a list of environment variables available for the job, run the job with the **Debug log file** option on. The debug log file lists all environment variables and, if applicable, their values. Environment variables are preceded by **env** in the log file.

7. Choose whether to set the **Debug log file** option. This option allows you to obtain diagnostic details to help debug job properties when setting up a job. You can set this option and use the **Run Now** feature on the **Job Summary** page. Once you have submitted the job, Fathom makes debug log file data available through the debug log file link on the **Job Summary** page.
8. Select the **Indicate if the job can be used as an action** check box if you want this job to appear in the list of actions that can be selected for execution from the **Alerts and Job completion action** page.
9. If you intend to set up actions and alerts for this job, click the **Edit** button associated with the **Completion Actions and Alerts** field to display the **Job Completion Actions and Alerts** page. The **Job Completion Actions and Alerts** page appears. See the [Resource Monitoring Guide](#) for more information about the **Job Completion Actions and Alerts** page.
10. Click **Save**. The **Job Summary** page appears. You can now schedule the job or run it immediately.

Advanced information

After you create a job instance, it is possible to view and edit advanced information about the job.

Note: The Truncate BI job and the Grow BI job do not allow you to view advanced information.



To edit the advanced information for a database maintenance job:

1. Click **View** on the **Job Summary** page.
2. Click **Edit**. The **Advanced information** page appears.
3. Enter the account information for the user who will run the job. Note that the default is the user who started the AdminServer.
4. If you wish to change the job specifications, enter the appropriate command and command parameters. If you want to change job specifications, consider creating a new job template. See the [Resource Monitoring Guide](#) for steps on creating custom job templates.

5. Enter applicable environment variables. See [Table 4–1](#) for a list of database maintenance job environment variables.
6. Click **Save**.

Environment variables

Use the information provided in [Table 4–1](#) if you choose to create custom jobs by modifying the provided job templates.

Table 4–1: Environmental variables for database maintenance jobs (1 of 4)

Job	Variable	Description
Database Backup	FM_BKPDEVICE	Identifies the backup device, either the tape or complete path to the file.
Database Backup	FM_BKPDEVICETYPE	Indicates the type of device to be used: Disk or Tape.
Database Backup	FM_BKPTAPEDIR	Identifies the full tape device name. This name can match the value defined for FM_BKPDEVICE.
Database Backup	FM_BKPDIRECTORY	Identifies the directory for file backup.
Database Backup	FM_BKPFILENAME	Identifies the filename for file backup.
Database Backup	FM_BKPOVERWRITEFILE	Indicates whether a file is overwritten. This information applies only to files, not tapes.
Database Backup	FM_BKPONLINE	Identifies the option that determines whether the backup job is to be performed online or offline. If this option is not set, Fathom assumes the backup is offline.

Table 4–1: Environmental variables for database maintenance jobs (2 of 4)

Job	Variable	Description
Database Backup	FM_BKPINCREMENTAL	Indicates whether data in blocks that have been changed since the last backup will be backed up.
Database Backup	FM_BKPESTIMATESIZE	Indicates whether a report that estimates the size of the backup will be generated.
Database Backup	FM_BKPVERBOSELISTING	Indicates whether Fathom will provide a verbose output every 10 seconds concerning the status of the backup.
Database Backup	FM_BKPCOMPRESSSION	Indicates whether you want data blocks compressed as they are backed up.
Database Backup	FM_BKPNORECOVERY	Used for mirrored backups. Set this option if you do not want the database to be able to roll forward an AI file.
Database Backup	FM_BKPVOLUMESIZE	Identifies the size that you want the backup to be before prompting for another volume.
Database Backup	FM_BKPBLOCKINGFACTOR	Specifies the blocking factor size used to flush buffers out of the backup device.
Database Backup	FM_BKPREUNDANCYFACTOR	Specifies the redundancy count to create redundancy in the backup.
Database Restore	FM_RSTDEVICE	Identifies the device to which the database is restored. It can be either a tape or a complete path to a file.

Table 4–1: Environmental variables for database maintenance jobs (3 of 4)

Job	Variable	Description
Database Restore	FM_RSTDEVICETYPE	Indicates the type of device to be used: Disk or Tape.
Database Restore	FM_RSTVERIFY	Identifies one of three possible restore options: <ul style="list-style-type: none"> Partial verification. Full verification. Creation of a structure file from the backup that was completed.
Database Restore	FM_RSTDBNAME	Identifies the name of the database you intend to restore. This name is the full path and name of the database you want to restore.
GrowBI	FM_BICLUSTERNUMBER	The number of clusters to pregrow the BI file to when the command is run. The initial size of the BI file is determined by taking the BI cluster size times the BICLUSTERNUMBER.
Database Analysis	FM_DBAUTILIZATIONNUMBER	Specifies the percentage of utilization of the blocks for the index to validate against.
Database Analysis	FM_COMPACTIONNUMBER	Specifies the percentage to which indexes will be packed when index compaction is run.
Database Analysis	FM_DBABLOCKNUMBER	Specifies the threshold for the number of blocks that the index must contain for the rule to be run.

Table 4–1: Environmental variables for database maintenance jobs *(4 of 4)*

Job	Variable	Description
Database Analysis	FM_DBARULEVALIDATION	Indicates that you want this job to test your rules.
Index Compaction	FM_IDXTABLENAME	Identifies the name of the table to compact.
Index Compaction	FM_IDXINDEXNAME	Identifies the name of the index to compact.
Index Compaction	FM_COMPACTIONNUMBER	Specifies the percentage to which indexes will be packed when index compaction is run.
All database maintenance jobs	FM_CONTAINERNAME	Specifies the resource's container and the resource. For example: Dev01.sports2000.

Index

A

Advanced Views 3–39
After Imaging Area view 3–25
Application Data Areas view 3–22

B

Before Imaging Area view 3–24

C

Control Area view 3–21

D

Database
 changing control settings 3–3 to 3–5
 Details page 3–2
 maintenance 4–2
 migrating 1–3
 remote 1–10
 rules 1–11
Database log file viewer, *See* Log file viewer
Database maintenance job templates, *See*

Job templates

Database migration utility 1–3

Database monitoring plans, *See* Monitoring plans

Database resource monitors
 defaults 1–5
 rules 1–11
 updating 1–7

Database Rules, *See* Rules

Database views
 Advanced 3–38 to 3–39
 File Systems 3–15 to 3–17
 Locks and Latches 3–31
 Memory Resources 3–33
 Page Writer 3–35 to 3–36
 Raw VST data 3–38
 Record and Index Activity 3–30
 Start Parameters 3–37
 Storage Area 3–18 to 3–25
 Transactions 3–32
 User Activity 3–26 to 3–28
 Vital Signs 3–13

F

File resource monitors
 creating 3–17

G

Graph caching 2–5

Graphs

- general information 2–4
- meter 2–3
- pinup 2–6 to 2–8

J

Job templates, *See* Templates

L

Log file monitors

- customizing 3–7 to 3–8
- general considerations 3–7

M

Monitoring plans

- default values 1–6
- modifying 1–7
- rule sets 1–16

R

Raw VST Data view 3–38

Record and Index Activity view 3–30

Resource monitoring plans, *See* Monitoring plans

Rule set

- adding 1–16
- associating 1–16
- database
 - copying 1–15
 - creating 1–14
 - deleting 1–16

Rules

- details 1–13

S

Schedules

- Default_Schedule 3–8

Start Parameters view 3–37

Storage Area view 3–18 to 3–20

System Areas view 3–21

T

Templates

- accessing 4–4
- backup jobs
 - unattended 4–5
- database analysis 4–14
- database index compaction 4–15
- database restore 4–8
- definition 4–2
- grow BI 4–12
- purpose 4–2
- truncate BI 4–10
- viewing 4–2

Transactions view 3–32

U

User Activity view 3–26 to 3–28

V

Views. *See* Database views

Vital Signs view 3–13