Progress®



OpenEdge® Management:
Trend Database Guide and Reference

OpenEdge®

Table of Contents

Preface....7
Purpose....9
Audience....11
Organization....13
Use ABL documentation....15
References to ABL compiler and run-time features....15
References to ABL data types....15
Typographical conventions....17
Examples of syntax descriptions....19
Long syntax descriptions split across lines....20
Complex syntax descriptions with both required and optional elements....21
OpenEdge messages....23
Obtain more information about OpenEdge messages....23

Managing OpenEdge Management Trend Database Data....25
Managing the OpenEdge Management Trend Database size....25
Sizing guidelines for the OpenEdge Management Trend Database....26
How data compacting works....27
Running compaction as a scheduled job....28
Compacting and purging data....28
Compacting and retaining data....28
Purging data....28

Database Compaction Job....29
Creating compaction job instances....29
Editing advanced information....31
Environment variables for Trend Data Compaction jobs....33
Viewing compaction statistics....33
Running the Data Compaction Unlock job instance....35

OpenEdge Management Trend Database Schema....37
Table and field name definitions....38
Configuration tables....38
Content of the configuration tables....39
Compaction details for configuration tables....43
OpenEdge Management Trend Database tables....43
Contents of the OpenEdge Management Trend Database tables....44

Compaction details for database tables....57
OpenEdge tables....59
System tables....64
Compaction details for system tables....68
Archive tables....69
Common archive fields....71
Complete contents of an archived table....72
Database analysis tables....73
Db_TabAnalysis table....73
Db_IdxAnalysis table....74
Network table....75

Index....77

Copyright

© 2020 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

These materials and all Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

Corticon, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, DataRPM, Defrag This, Deliver More Than Expected, Icenium, Ipswitch, iMacros, Kendo UI, Kinvey, MessageWay, MOVEit, NativeChat, NativeScript, OpenEdge, Powered by Progress, Progress, Progress Software Developers Network, SequeLink, Sitefinity (and Design), Sitefinity, SpeedScript, Stylus Studio, TeamPulse, Telerik, Telerik (and Design), Test Studio, WebSpeed, WhatsConfigured, WhatsConnected, WhatsUp, and WS_FTP are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries. Analytics360, AppServer, BusinessEdge, DataDirect Autonomous REST Connector, DataDirect Spy, SupportLink, DevCraft, Fiddler, iMail, JustAssembly, JustDecompile, JustMock, NativeScript Sidekick, OpenAccess, ProDataSet, Progress Results, Progress Software, ProVision, PSE Pro, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, and WebClient are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Any other marks contained herein may be trademarks of their respective owners.

March 2020

Product version: Progress OpenEdge 12.2

Preface

Purpose

This guide describes how to manage your OpenEdge® Management Trend Database. This guide assumes you are familiar with basic OpenEdge Management monitoring terms and concepts, as explained in *OpenEdge Management: Resource Monitoring*.

Audience

This guide is designed for users of the OpenEdge Management product. Typical users are OpenEdge database administrators and any others responsible for the daily management of the OpenEdge Management Trend Database.

Organization

Managing OpenEdge Management Trend Database Data on page 25

Provides an overview of ways to manage OpenEdge Management Trend Database data, including data compaction and purging. Also gives examples of how the size of the OpenEdge Management Trend Database grows over time.

Database Compaction Job on page 29

Provides details about creating and running a Database Compaction job instance and, if necessary, a Data Compaction Unlock job instance.

OpenEdge Management Trend Database Schema on page 37

Describes the OpenEdge Management Trend Database schema.

Use ABL documentation

OpenEdge provides a special purpose programming language for building business applications. In the documentation, the formal name for this language is *ABL (Advanced Business Language)*. With few exceptions, all keywords of the language appear in all UPPERCASE, using a font that is appropriate to the context. All other alphabetic language content appears in mixed case.

For the latest documentation updates go to https://docs.progress.com.

References to ABL compiler and run-time features

ABL is both a compiled and an interpreted language that executes in a run-time engine. The documentation refers to this run-time engine as the *ABL Virtual Machine (AVM)*. When the documentation refers to ABL source code compilation, it specifies *ABL* or *the compiler* as the actor that manages compile-time features of the language. When the documentation refers to run-time behavior in an executing ABL program, it specifies *the AVM* as the actor that manages the specified run-time behavior in the program.

For example, these sentences refer to the ABL compiler's allowance for parameter passing and the AVM's possible response to that parameter passing at run time: "ABL allows you to pass a dynamic temp-table handle as a static temp-table parameter of a method. However, if at run time the passed dynamic temp-table schema does not match the schema of the static temp-table parameter, the AVM raises an error." The following sentence refers to run-time actions that the AVM can perform using a particular ABL feature: "The ABL socket object handle allows the AVM to connect with other ABL and non-ABL sessions using TCP/IP sockets."

References to ABL data types

ABL provides built-in data types, built-in class data types, and user-defined class data types. References to built-in data types follow these rules:

- Like most other keywords, references to specific built-in data types appear in all UPPERCASE, using a font that is appropriate to the context. No uppercase reference ever includes or implies any data type other than itself.
- Wherever *integer* appears, this is a reference to the `INTEGER` or `INT64` data type.
- Wherever *character* appears, this is a reference to the `CHARACTER`, `LONGCHAR`, or `CLOB` data type.
- Wherever *decimal* appears, this is a reference to the `DECIMAL` data type.
- Wherever *numeric* appears, this is a reference to the `INTEGER`, `INT64`, or `DECIMAL` data type.

References to built-in class data types appear in mixed case with initial caps, for example, `Progress.Lang.Object`. References to user-defined class data types appear in mixed case, as specified for a given application example.

Typographical conventions

This documentation uses the following typographical and syntax conventions:

Convention	Description
Bold	Bold typeface indicates commands or characters the user types, provides emphasis, or the names of user interface elements.
Italic	Italic typeface indicates the title of a document, 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 plus sign between key names indicates a **simultaneous** key sequence: you press and hold down the first key while pressing the second key. For example, **CTRL+X**.
KEY1 KEY2	A space between key names indicates a **sequential** 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, code examples, system output, and file names.
`Fixed-width italics`	Fixed-width italics indicate variables in syntax.
`Fixed-width bold`	Fixed-width bold italic indicates variables in syntax with special emphasis.
`UPPERCASE fixed width`	ABL keywords in syntax and code examples are almost always shown in upper case. Although shown in uppercase, you can type ABL keywords in either uppercase or lowercase in a procedure or class.
Period (.) or colon (:)	All statements except `DO`, `FOR`, `FUNCTION`, `PROCEDURE`, and `REPEAT` end with a period. `DO`, `FOR`, `FUNCTION`, `PROCEDURE`, and `REPEAT` statements can end with either a period or a colon.
[ ]	Large brackets indicate the items within them are optional.
[ ]	Small brackets are part of ABL.
{ }	Large braces indicate the items within them are required. They are used to simplify complex syntax diagrams.
{ }	Small braces are part of ABL. For example, a called external procedure must use braces when referencing arguments passed by a calling procedure.

Convention	Description
|	A vertical bar indicates a choice.
. . .	Ellipses indicate repetition: you can choose one or more of the preceding items.

Examples of syntax descriptions

In this example, ACCUM is a keyword, and *aggregate* and *expression* are variables:

Syntax

```
ACCUM aggregate expression
```

FOR is one of the statements that can end with either a period or a colon, as in this example:

```
FOR EACH Customer NO-LOCK:
  DISPLAY Customer.Name.
END.
```

In this example, STREAM *stream*, UNLESS-HIDDEN, and NO-ERROR are optional:

Syntax

```
DISPLAY [ STREAM stream ] [ UNLESS-HIDDEN ] [ NO-ERROR ]
```

In this example, the outer (small) brackets are part of the language, and the inner (large) brackets denote an optional item:

Syntax

```
INITIAL [ constant [ , constant ] ]
```

A called external procedure must use braces when referencing compile-time arguments passed by a calling procedure, as shown in this example:

Syntax

```
{ &argument-name }
```

In this example, EACH, FIRST, and LAST are optional, but you can choose only one of them:

Syntax

```
PRESELECT [ EACH | FIRST | LAST ] record-phrase
```

In this example, you must include two expressions, and optionally you can include more. Multiple expressions are separated by commas:

Syntax

```
MAXIMUM ( expression , expression [ , expression ] ... )
```

In this example, you must specify MESSAGE and at least one *expression* or SKIP [(*n*)], and any number of additional *expression* or SKIP [(*n*)] is allowed:

Syntax

```
MESSAGE { expression | SKIP [ ( n ) ] } ...
```

In this example, you must specify {*include-file*, then optionally any number of *argument* or &*argument-name* = "*argument-value*", and then terminate with }:

Syntax

```
{ include-file
    [ argument | &argument-name = "argument-value" ] ... }
```

Long syntax descriptions split across lines

Some syntax descriptions are too long to fit on one line. When syntax descriptions are split across multiple lines, groups of optional and groups of required items are kept together in the required order.

In this example, WITH is followed by six optional items:

Syntax

```
WITH [ ACCUM max-length ] [ expression DOWN ]
  [ CENTERED ] [ n COLUMNS ] [ SIDE-LABELS ]
  [ STREAM-IO ]
```

Complex syntax descriptions with both required and optional elements

Some syntax descriptions are too complex to distinguish required and optional elements by bracketing only the optional elements. For such syntax, the descriptions include both braces (for required elements) and brackets (for optional elements).

In this example, ASSIGN requires either one or more *field* entries or one *record*. Options available with *field* or *record* are grouped with braces and brackets:

Syntax

```
ASSIGN   { [ FRAME frame ] { field [ = expression ] }
           [ WHEN expression ] } ...
       | { record [ EXCEPT field ... ] }
```

OpenEdge messages

OpenEdge displays several types of messages to inform you of routine and unusual occurrences:

- **Execution messages** inform you of errors encountered while OpenEdge is running a procedure; for example, if OpenEdge cannot find a record with a specified index field value.
- **Compile messages** inform you of errors found while OpenEdge is reading and analyzing a procedure before running it; for example, if a procedure references a table name that is not defined in the database.
- **Startup messages** inform you of unusual conditions detected while OpenEdge is getting ready to execute; for example, if you entered an invalid startup parameter.

After displaying a message, OpenEdge proceeds in one of several ways:

- Continues execution, subject to the error-processing actions that you specify or that are assumed as part of the procedure. This is the most common action taken after execution messages.
- Returns to the Procedure Editor, so you can correct an error in a procedure. This is the usual action taken after compiler messages.
- Halts processing of a procedure and returns immediately to the Procedure Editor. This does not happen often.
- Terminates the current session.

OpenEdge messages end with a message number in parentheses. In this example, the message number is `200`:

```
** Unknown table name table. (200)
```

If you encounter an error that terminates OpenEdge, note the message number before restarting.

Obtain more information about OpenEdge messages

In Windows platforms, use OpenEdge online help to obtain more information about OpenEdge messages. Many OpenEdge tools include the following Help menu options to provide information about messages:

- Choose **Help** > **Recent Messages** to display detailed descriptions of the most recent OpenEdge message and all other messages returned in the current session.
- Choose **Help** > **Messages** and then type the message number to display a description of a specific OpenEdge message.
- In the Procedure Editor, press the **HELP** key or **F1**.

On UNIX platforms, use the OpenEdge `pro` command to start a single-user mode character OpenEdge client session and view a brief description of a message by providing its number.

To use the pro command to obtain a message description by message number:

1. Start the Procedure Editor:

```
OpenEdge-install-dir/bin/pro
```

2. Press **F3** to access the menu bar, then choose **Help** > **Messages**.
3. Type the message number and press **ENTER**. Details about that message number appear.
4. Press **F4** to close the message, press **F3** to access the Procedure Editor menu, and choose **File** > **Exit**.

1

Managing OpenEdge Management Trend Database Data

The amount of data stored in the OpenEdge® Management Trend Database expands with usage. You can perform compaction and purging of this data with the compaction and cleanup process.

For details, see the following topics:

- Managing the OpenEdge Management Trend Database size
- How data compacting works
- Running compaction as a scheduled job

Managing the OpenEdge Management Trend Database size

Because the OpenEdge Management Trend Database stores all the trend data, it can become too large and require maintenance. You have several options in managing the data stored in the database, such as:

- Compacting the data, moving it from the OpenEdge Management Trend Database into an archive, and then purging the archive
- Compacting the data, moving it from the OpenEdge Management Trend Database into an archive, and retaining the archive
- Purging the data without compacting it

Sizing guidelines for the OpenEdge Management Trend Database

As OpenEdge Management runs, it gathers statistics on defined resources. These statistics are accumulated in several tables in the OpenEdge Management Trend Database. Every time OpenEdge Management trends data from a resource, a sample is added to that particular table. The following example shows how the amount of data trended affects the size of the OpenEdge Management Trend Database.

Example

For this example, the following quantities of trends are used for one trend sample:

- 5 alerts
- 10 tasks (jobs and/or reports)
- 25 database files
- 10 areas
- 10 checkpoints
- 25 system disks
- 25 file systems
- 5 FileSize
- 50 Database Analysis
- 50 Index Analysis

The following are single rows in the OpenEdge Management Trend Database:

- Samples
- Activity for APW
- Buffers
- Index
- I/O by type
- Locks
- Logging (AI and BI)
- Records
- Servers
- Summary
- Network
- CPU
- Memory
- Status

Note: If Table Statistics and Index Statistics are trended, there are 50 rows of each.

The sample resources will render approximately 12.2K worth of data stored in the OpenEdge Management Trend Database. This data is broken down as follows:

- Database — 3.4K
- System information — 4.1K
- Configuration information (including tasks) — 3.0K
- Index information (for all of the above samples) — 1.7K

If Table Statistics and Index Statistics are trended, they result in 5.4K and 7.1K worth of data, respectively. Additional index information for Table and Index Statistics rows results in approximately 920 bytes.

Scenario one

Using the quantities and values defined for the example, trending every 5 minutes for 24 hours over a 7-day period results in 2016 samples (288 5-minute periods in a day multiplied by 7 days). These 2016 samples result in an OpenEdge Management Trend Database of approximately 28.3MB.

Scenario two

Using the quantities and values defined for the example, trending every 15 minutes for 24 hours over a 7-day period results in 672 samples (96 15-minute periods in a day multiplied by 7 days). These 672 samples result in an OpenEdge Management Trend Database of approximately 22.5MB.

How data compacting works

When you run OpenEdge Management Trend Database compaction (as a job), the compaction takes a number of the records and compacts them in a different table in the OpenEdge Management Trend Database. For example, the compaction might take the first four records in the table and combine them into one record in the archive table.

As part of the compaction process, the records physically move to the archive table and are deleted from the real-data table. The archive automatically creates an hourly, daily, and weekly version of each sample, as follows:

- **Hourly** — Takes all the data for an hour and compacts it into one record for that hour. This compaction method takes records for one hour and reduces them to one row.
- **Daily** — Takes all the data for a day and compacts it into one record for that day. This compaction method takes records for one day and reduces them to one row.
- **Weekly** — Takes all the data for seven days and compacts it into one record. The week technically begins at midnight on Saturday night. This compaction method takes records for seven days and reduces them to one row.

Caution: There is no option to undo the deletion of data samples from a real-data table following the compaction.

Running compaction as a scheduled job

OpenEdge Management provides a default Database Maintenance job that you can use to run the compaction. This default job runs the compaction once each week on any data older than one month. By compacting only the most historical data, you can save some of the current data in the table and use it for more detailed reporting purposes.

You can determine at what time the compaction is run, and you can also change the job characteristics if you want to run the compaction more or less frequently. See Creating compaction job instances on page 29 for detailed steps on setting up a compaction job instance.

Compacting and purging data

Specifying a different compaction date instead of purging the data also helps to maintain your database size. You should compact data that is older than 30 days and purge data that is older than 540 days.

Compacting and retaining data

When you run the compacting job, the compaction moves the database records into an archive table. You can retain the archive table and still have access to the compacted data, but after the compaction, the data is available only in the archive table. The compaction process removes the data from the real-data table.

Purging data

If you want to purge data without compacting it, you can specify the same number of days for the compacting and the purging.

2

Database Compaction Job

Use the Data Compaction job to manage the size of your OpenEdge Management Trend Database. This chapter explains how to create and schedule a data compaction job instance, how to view a finished job's statistics, and how to run Data Compaction again if the job instance terminates unexpectedly.

For details, see the following topics:

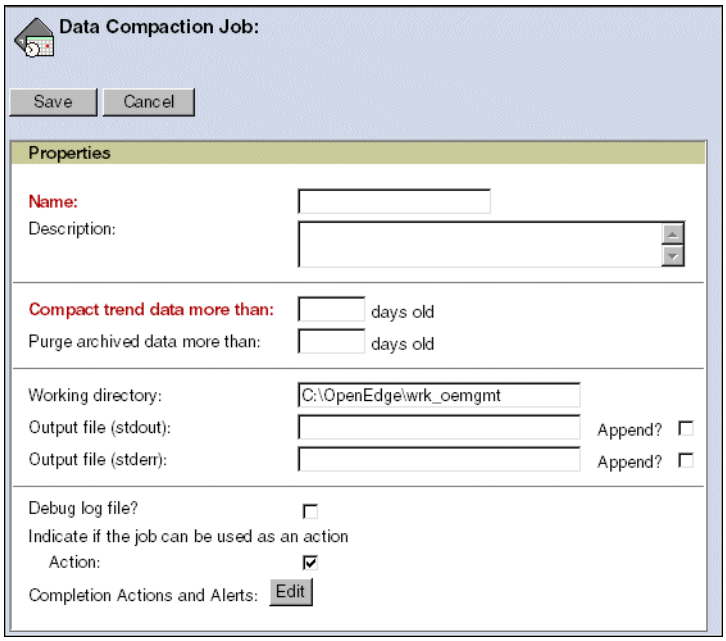
- Creating compaction job instances
- Editing advanced information
- Viewing compaction statistics
- Running the Data Compaction Unlock job instance

Creating compaction job instances

The Data Compaction job compacts and purges data in the OpenEdge Management Trend Database. The primary purpose of this job is to help you to manage the size of the database.

To access the job, choose **Jobs** > **New** > **Job from a Template**. Then choose the **DataCompaction** link on the **Create Custom Job** page. The **Data Compaction Job** page appears, as shown in the following figure.

Figure 1: Data Compaction Job page



You must complete the necessary fields on the page to set up and run a database compaction job instance.

To set up and run a data compaction job:

1. Provide values for the **Name** and the **Description** fields. **Name** is a required field.
2. In the **Compact trend data more than days old** field, identify the number of days for which you intend to compact data. A day, as defined for this field, begins at midnight. This is a required field.

   Note: You can compact only data that is more than a week old. If you enter a value of 1-6, no data compaction will take place.

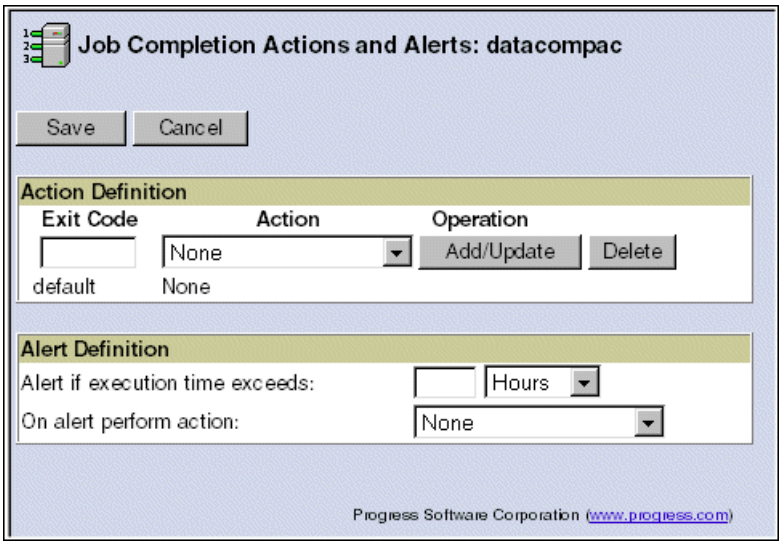
   In the **Purge archived data more than days old** field, specify the number of days of data you want purged from the database.

3. Enter a value in the **Working directory** field. The **Working directory** field identifies the command's current working directory, and the directory you specify must exist. This property defaults to the working directory defined at installation.
4. Enter values in the **Output file (stdout)** and **Output file (stderr)** fields. The **stdout** location indicates where messages will be stored and the **stderr** location indicates where error messages will be stored. To append subsequent messages to these files, select the **Append** option.
5. Select the **Debug log file** option to obtain diagnostic details that help you 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, OpenEdge Management makes debug trace file data available through the debug trace file link in the **Job History** section of the **Job Summary** page.

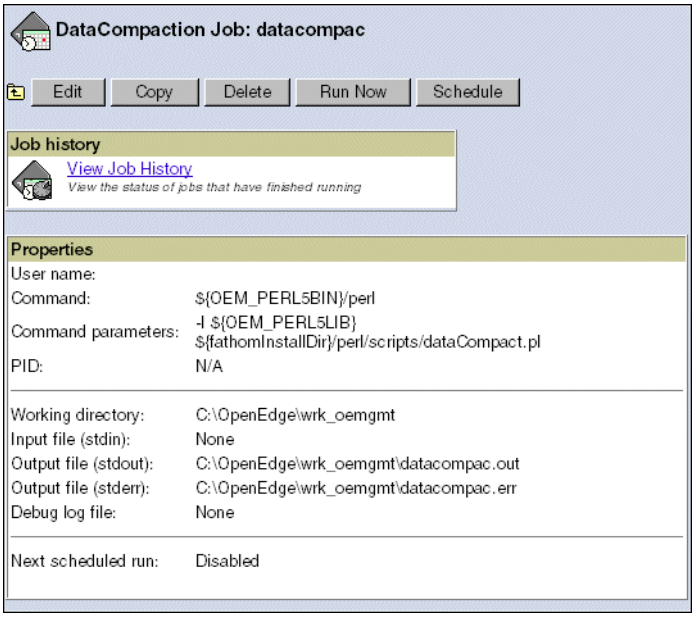
6. Select the **Action** option to indicate if the job can be used as an action. See *OpenEdge Management: Resource Monitoring* for more information about actions.

   If you intend to set up actions and alerts for this job, click the **Edit** button associated with the **Completion Actions and Alerts** field. The **Job Completion Actions and Alerts** page appears:



7. Define any actions or alerts, then click **Save** to return to the **Data Compaction Job** page. See *OpenEdge Management: Resource Monitoring* for more information about associating actions with alerts.

   The summary page appears:



   Once you create a Data Compaction job instance, you can view and edit the job's advanced information.

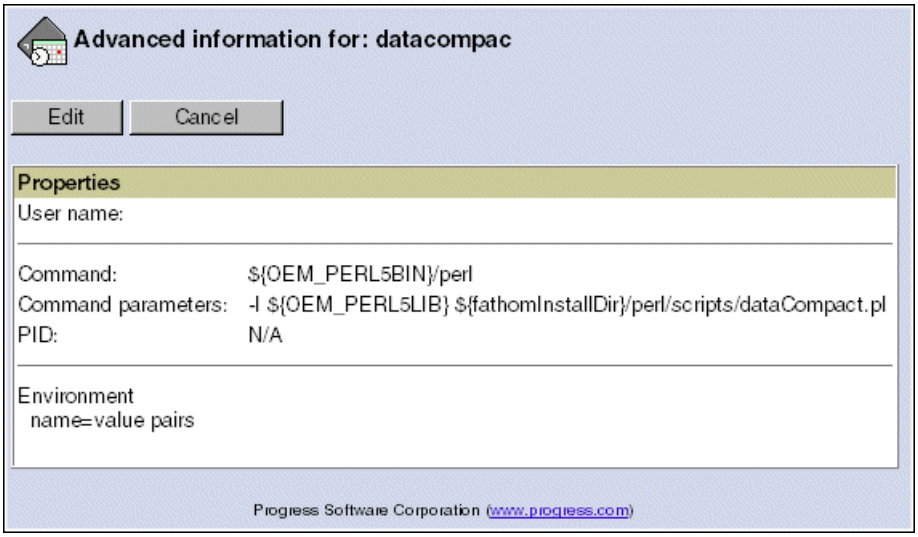
Editing advanced information

The **Advanced information for** page displays the name of the user, the job's command, the command's parameters, and the job's environment variables.

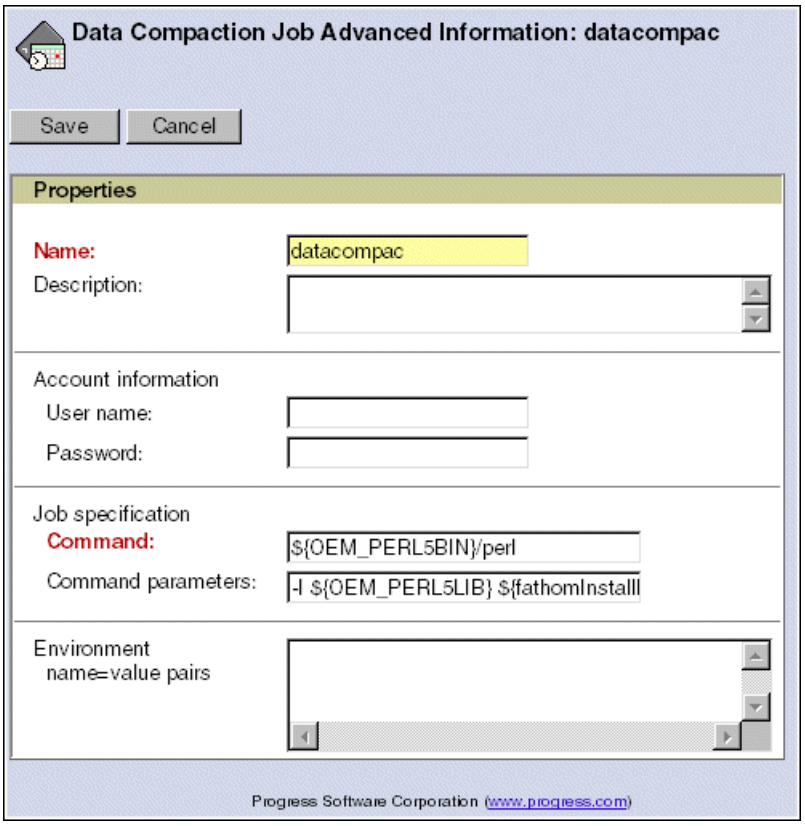
To edit the advanced information for a data compaction job:

1. From the list frame, expand **Defined Jobs**.
2. Click the data compaction job.

3. Click **view** (next to the **Advanced information** entry) on the **Data Compaction Job** summary page. The **Advanced information for** page appears:



4. Click **Edit**. The **Data Compaction Job Advanced Information** page appears:



5. Provide the account information for the user who will run the job. Note that the default is the user who started the AdminServer.
6. To change the job specifications, enter the appropriate command and command parameters. If you want to change the job specifications, consider creating a new job template. See *OpenEdge Management: Resource Monitoring* for the steps to create custom job templates.

7. Enter applicable environment variables. See the Environment variables for Trend Data Compaction jobs table for Trend Data Compaction environment variables.
8. Click **Save**. The **Advanced Information** page reappears.

Environment variables for Trend Data Compaction jobs

The following table identifies and describes the environment variables used with database compaction jobs.

Table 1: Environment variables for Trend Data Compaction jobs

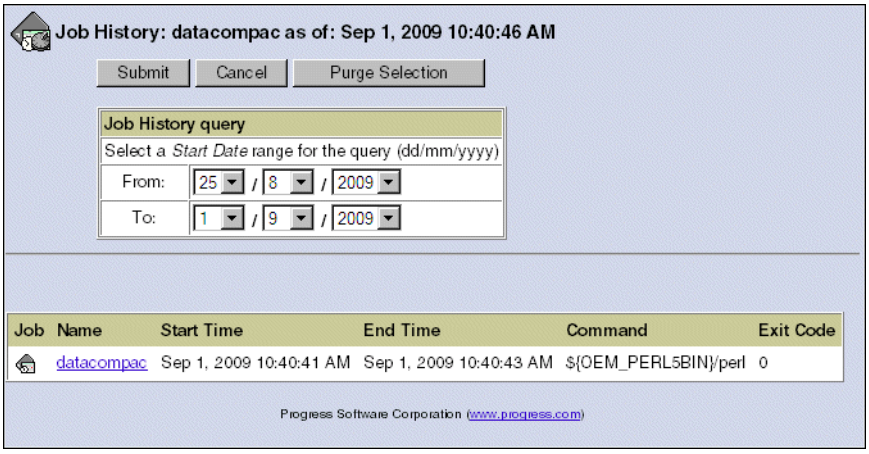
Variable	Description
FM_DBCCOMPACTDAYS	Identifies the number of days for which you intend to compact data. A day, as defined for this field, begins at midnight.
FM_DBCPURGEDAYS	Specifies the number of days' worth of data you want purged from the database.

Viewing compaction statistics

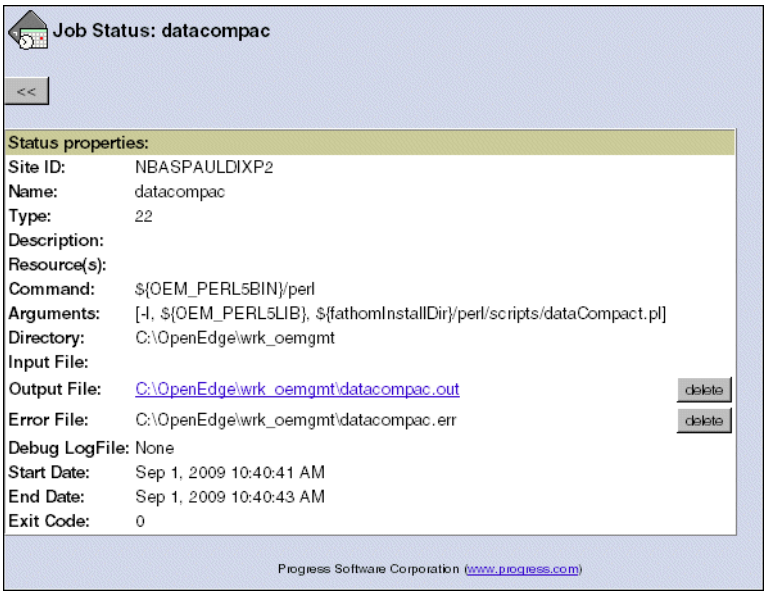
You can review information regarding data compaction by looking in the output file you specified when you set up the compaction job.

To access a job's output files:

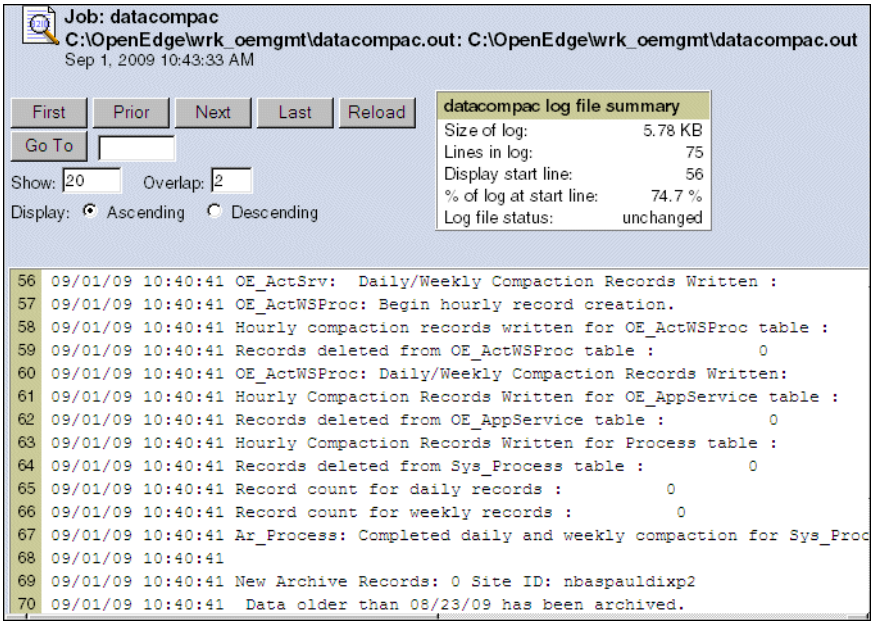
1. From the list frame, expand **Defined Jobs**.
2. Click the data compaction job.
3. Click **View Job History**. The **Job History** page appears.
4. Submit the date range for your query. Jobs that ran during the time period specified appear at the bottom of the **Job History** page, as shown:



5. Click the name of the job. In this example, the job is **datacompac**. The **Job Status** page appears:



6. Click the link to the <*name*>`.out` file. The file displays in a log file viewer:



The following information helps you navigate through the job output and error files:

- Use the **Show** field to define how many database log file entries will display at one time. The number entered into the **Show** field cannot be less than 10.
- Use the **Overlap** field to define how many entries are repeated from screen to screen 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 continues to display the previous values.
- Click **Go To** to define the numbered entry in the log file with which the viewer will begin its display. 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 does not update its display.

- The default display of entries is in ascending order; choose **Descending** to change the display.
- 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**.

OpenEdge Management considers a viewer that has been inactive for more than four hours stale. Once a viewer becomes stale, OpenEdge Management releases ninety-five percent of any memory being held. If you try to use a stale viewer, OpenEdge Management 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.

OpenEdge Management considers a viewer that has been inactive for forty-eight hours dead. Once a viewer is dead, OpenEdge Management releases all of its memory. To return to the log file displayed in a dead view, you must renavigate to it, even if you pinned up the view or saved a link to it before the viewer died.

Running the Data Compaction Unlock job instance

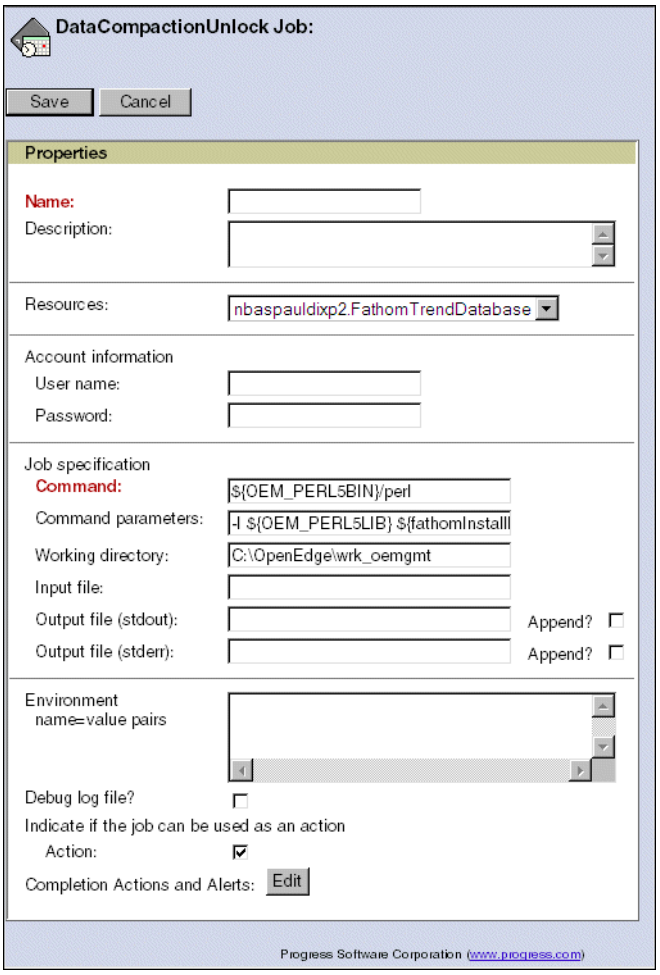
When you start the Data Compaction job, a flag is inserted into the Cf_Control table in the OpenEdge Management Trend Database. The flag indicates that the job is currently running and prevents the job from being run twice. When the job ends successfully, the flag is reset, which allows you to run the job again in the future.

If, however, the Data Compaction job terminates unexpectedly before it completes (for example, if the machine on which it is running is rebooted, or if someone kills the job), the inserted flag is not automatically reset. To reset the flag in such a situation, run the Data Compaction Unlock job.

To run the data compaction unlock job:

1. Access the job by choosing **Jobs** > **New** > **Job from a Template**. Then click **DataCompactionUnlock** on the **Create Custom Job** page.

   The **DataCompactionUnlock Job** page appears, as shown:



2. Under **Properties**, provide the values for the job.
3. Click **Save**. The **DataCompactionUnlockJob: <***job name***>** page appears.
4. Click **Run Now** to start the job immediately or **Schedule** to start the job at some other time.

   For more information about setting job properties, running jobs immediately, or running jobs by schedule, see *OpenEdge Management: Resource Monitoring*.

3

OpenEdge Management Trend Database Schema

When writing reports or viewing alerts, you might find it helpful to understand the OpenEdge Management database schema.

For details, see the following topics:

- Table and field name definitions
- Configuration tables
- OpenEdge Management Trend Database tables
- OpenEdge tables
- System tables
- Archive tables
- Database analysis tables
- Network table

Table and field name definitions

The naming of tables and fields in the OpenEdge Management database follows certain conventions. Each table and field name consists of a prefix and a suffix, separated by an underscore. The prefix indicates which resource the table or field is related to. The suffix indicates the name of the data and might contain a subsection. The suffix cannot be more than 15 characters long. The following table describes each prefix.

Table 2: Table prefix definitions

Prefix	Table or field contents	Description
Cf_	Configuration data	Tables or fields that store configuration information for such OpenEdge Management functions as schedules, resource identification, and alert definitions. See Configuration tables on page 38 for more information.
Sys_	System data	Tables or fields that store raw data collected from samplings of such resources as CPU, disk, and memory. See System tables on page 64 for more information.
Db_	Database data	Tables or fields that store data gathered from OpenEdge® databases. Sample data includes I/O, user statistics, and memory management. See Compaction details for configuration tables on page 43 and Database analysis tables on page 73 for more information.
OE_	OpenEdge Server data	Tables or fields that store data gathered from WebSpeed® and AppServer. **Note:** OpenEdge Management supports management of the WebSpeed® Transaction Server product. Throughout this guide, WebSpeed Transaction Server and WebSpeed are used interchangeably.
Ar_	Archive data	Tables or fields that store data gathered from OpenEdge databases for archiving purposes. Sample data includes all fields and data associated with a given table and archive-specific fields. Data stored in system and database tables can be archived; data stored in configuration tables cannot be archived. See the Archive tables on page 69 for more information.
Net_	Network data	Table or fields that store data gathered from TCP, UDP, ICMP, or HTTP resources.

Configuration tables

The schema of configuration tables is dependent upon many subsystems.

Content of the configuration tables

The following table provides a listing of each configuration table's contents.

Table 3: Configuration table contents

Table name	Table contents	Description
Control	Control information	Stores data about version information and counters
Resrc	Resource identification	Stores data about each resource that can be viewed and/or monitored
Sample	Sample master table	Stores data about any created samples, regardless of the resource
Site	Site master table	Stores data about each configured site
Status	Resource table status	Stores historical status changes of resources
Task_Detail	Task execution history	Stores data about each executed task
Task_Resrc	Task resource details	Stores data about task resources

The following table lists field names and data types for each configuration table. For more information about the filename prefix, see the Table prefix definitions table.

Table 4: Configuration table details

Filename	Field name	Data type	Description
Cf_Alert_Detail	Alert_ID	Integer	Sequential identification number of the alert
Cf_Alert_Detail	Resrc_ID	Integer	Standard resource identification number
Cf_Alert_Detail	Site_ID	Character	ID for the site that will contain resources being monitored; typically, the host name
Cf_Alert_Detail	Alert_Date	Character	Standard site identification number
Cf_Alert_Detail	Alert_Time	Integer	Date when the alert was generated
Cf_Alert_Detail	Alert_Message	Character	Message generated for the alert
Cf_Alert_Detail	Alert_Severity	Integer	Severity of the alert
Cf_Alert_Detail	Alert_Threshold	Decimal	Threshold value that was broken
Cf_Alert_Detail	Alert_Value	Decimal	Value that broke the threshold

Filename	Field name	Data type	Description
Cf_Alert_Detail	Alert_Who	Character	Name of the person who generated the alert
Cf_Alert_Detail	Alert_Name	Character	Name of the alert
Cf_Alert_Detail	Alert_Number	Integer	Number of the alert (that corresponds to the `promsgs` number)
Cf_Alert_Detail	Alert_ClearDate	Date	Date on which the alert was cleared
Cf_Alert_Detail	Alert_ClearWho	Character	Name of the person who cleared the alert
Cf_Alert_Detail	Alert_User1	Character	User-definable field
Cf_Alert_Detail	Alert_ClearTime	Integer	Time when the alert was cleared
Cf_Alert_Detail	Alert_ClearNote	Character	Specific note for the cleared alert
Cf_Resrc	Resrc_Desc	Character	Description of the resource
Cf_Resrc	Resrc_Host	Character	Fully qualified domain name of the resource's host machine
Cf_Resrc	Resrc_ID	Integer	Sequential identification number of the resource
Cf_Resrc	Resrc_IP	Character	IP address of the resource
Cf_Resrc	Resrc_Loc	Character	Location of the resource
Cf_Resrc	Resrc_Loc_Type	Character	Location type (directory, URL, or port)
Cf_Resrc	Resrc_Name	Character	Name of the resource
Cf_Resrc	Resrc_Port	Integer	Port number of the resource
Cf_Resrc	Resrc_Type	Character	Type of resource to be monitored
Cf_Resrc	Site_ID	Character	Identification for the site containing the resource
Cf_Sample	Resrc_ID	Integer	Sequential identification number of the resource
Cf_Sample	Sample_Date	Date	Date when the sample interval was taken
Cf_Sample	Sample_ID	Integer	Sequential identification number of the sample

Filename	Field name	Data type	Description
Cf_Sample	Sample_Len	Integer	Length, in seconds, of the sample interval
Cf_Sample	Sample_Time	Integer	Time when the sample was gathered
Cf_Sample	Sample_Trans	Integer	Number of transactions committed during the sample interval
Cf_Sample	Sample_Type	Character	Indicates the type of resource sampled. Possible values are: • 1 - Unknown • 3 - File Monitor • 5- TCP Port • 6 - UDP Port • 8 - Web Site • 11 - Database • 13 - DNS Device • 15 - FTP Site • 16 - ICMP Port • 19 - Log File • 22- Job • 23 - CPU • 24 - Memory • 25 - Disk • 26 - File System • 27 - User • 28 - Email Action • 29 - Command Action • 30 - Log Action • 31 - Report • 32 - Search String • 33 - Compound Action
Cf_Sample	Site_ID	Character	The ID given to the site (usually the host name)
Cf_Site	Site_ID	Character	Identification number for the site containing the resource
Cf_Site	Site_PRGS_Vers	Character	OpenEdge release number

Filename	Field name	Data type	Description
Cf_Site	Site_Serial	Integer	OpenEdge serial number
Cf_Site	Site_MachName	Character	Machine name of site machine
Cf_Site	Site_IP	Character	IP address of the site
Cf_Status	Resrc_ID	Integer	Resource ID of the database for this site
Cf_Status	Site_ID	Character	Site ID for the database
Cf_Status	Status_ID	Integer	Sequential number generated by OpenEdge Management
Cf_Status	Status_Date	Date	Date that the status change occurred
Cf_Status	Status_Time	Integer	Time that the status change occurred
Cf_Status	Status_User1	Character	User-definable field
Cf_Status	Status_Status	Character	The new status in effect
Cf_Task_Detail	Site_ID	Character	Standard site identification number
Cf_Task_Detail	Task_ID	Integer	Sequential identification number of the task that is used for storage
Cf_Task_Detail	Task_Type	Integer	Type of task executed (**22** indicates a user-defined task, and **31** indicates an OpenEdge Management report)
Cf_Task_Detail	Task_Name	Character	Name of the task
Cf_Task_Detail	Task_Desc	Character	Description of the task
Cf_Task_Detail	Task_Command	Character	Command for the task
Cf_Task_Detail	Task_Args	Character	Arguments passed to the task
Cf_Task_Detail	Task_WorkDir	Character	Working directory for the task
Cf_Task_Detail	Task_InFile	Character	Input file for the task; this file, if needed, is redirected into the command
Cf_Task_Detail	Task_OutFile	Character	Standard output file for the task
Cf_Task_Detail	Task_ErrFile	Character	Standard error file for the task
Cf_Task_Detail	Task_StartDate	Date	Start date for the task

Filename	Field name	Data type	Description
Cf_Task_Detail	Task_StartTime	Integer	Start time for the task
Cf_Task_Detail	Task_EndDate	Date	End date for the task
Cf_Task_Detail	Task_EndTime	Integer	End time for the task
Cf_Task_Detail	Task_ExitCode	Integer	Exit code passed back to OpenEdge Management: • **0** — Success • **Non-zero** — Failure
Cf_Task_Detail	Alert_ID	Integer	The alert ID
Cf_Task_Resrc	Task_ID	Integer	The sequential Task ID of this sample
Cf_Task_Resrc	Resrc_ID	Integer	The sequential Resource ID

Compaction details for configuration tables

The following table summarizes how the database compaction process affects data rows contained in the configuration tables.

Table 5: Configuration tables and the compaction process

Configuration table	What happens to table data during compaction
Cf_Sample	Sample rows will be removed and renumbered.
Cf_Alert_Detail	There is no data in this table to be summed. The contents of this table will be purged.
Cf_Task_Detail	There is no data in this table to be summed. The contents of this table will be purged.

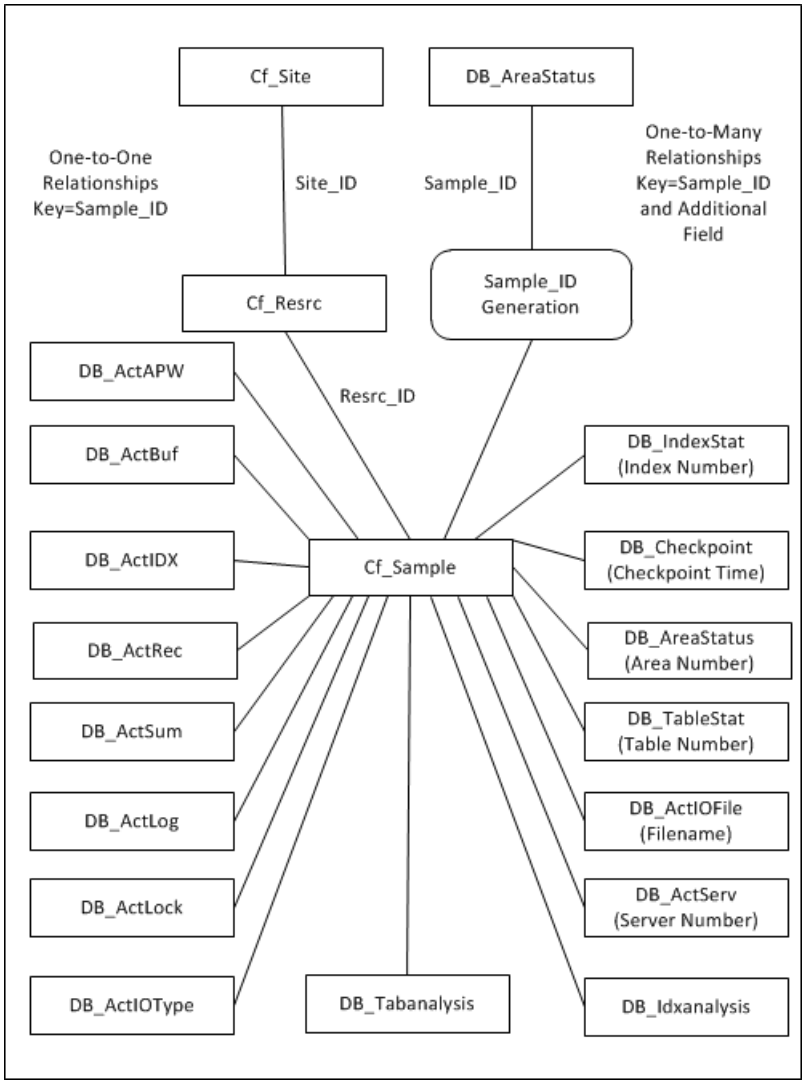
OpenEdge Management Trend Database tables

If you have used the PROMON utility with an OpenEdge database before, the information displayed in the OpenEdge Management Trend Database tables might look familiar to you. However, there is a difference between the information available through OpenEdge Management tables and the information through PROMON.

OpenEdge Management tables store data in one place, making it faster for you to view data. For example, all user information is contained in one table. To access user information with PROMON, you would have to view more than one PROMON screen. Unlike PROMON, OpenEdge Management must access only one table to return a view of user statistics.

As with configuration tables, the schema of these tables is dependent upon many subsystems. The following table shows the relationship among these tables and describes the data gathered for each trended database. Every DB_ table contains the sample's ID number. The sample table stores data from the gathered sample, which applies to all the tables. The sample table also contains the sample's Site and Resource IDs. A control record in the control table generates the Sample ID.

Figure 2: Database table relationships



Contents of the OpenEdge Management Trend Database tables

The following table describes each table and identifies from which VST the information is gathered.

Table 6: OpenEdge Management Trend Database tables

Table name	Table contents	Description
ActAPW	Information from the _ActPWs VST	Stores APW activity data, such as the number of buffer reads and writes

Table name	Table contents	Description
ActBuf	Information from the _ActBuffer VST	Stores buffer activity data, such as the number of marked and flushed buffers, and alternate buffer pool values (for Enterprise RDBMS only)
ActIdx	Information from the _Index and _ActIndex VSTs	Stores index data about overall performance
ActIOFile	Information from the _ActIOFile VST	Stores I/O data about each database file, except for the `.lk` and `.lg` files
ActIOType	Information from the _ActIOType VST	Stores database I/O data, such as the number of BI and AI blocks read
ActLock	Information from the _ActLock and _LockReq VSTs	Stores lock activity data, such as the number of requests and the number of database locks
ActLog	Information from the _ActAILog and _ActBILog VSTs	Stores activity data about before- and after-imaging
ActRec	Information from the _ActRecord and _ActSummary VSTs	Stores performance data about record reads, writes, and updates
ActServ	Information from the _ActServer and _Server VSTs	Stores server process data, such as bytes and records sent and received and the number of users
AreaStatus	Information from the _AreaStatus VST	Stores data about each area's HWM, Free Chain, and RM Chain
ActSum	Information from the _ActSummary VST	Stores database activity data, such as the number of records created and deleted
Checkpoint	Information from the _Checkpoint VST	Stores data about checkpoint operations
IndexStat	Information from the _IndexStat and _Index VSTs	Stores individual index data about reads and writes

The following table details each schema table, including fields, data types, a brief description, and how data for each field is handled when data compaction activities occur. For more information about each filename prefix, see the Table prefix definitions table.

Table 7: OpenEdge Management Trend Database table details

Filename	Field name	Data type	Description	Compaction status
Db_ActAPW	ActAPW_APWQWrites	Decimal	Number of buffer writes from the APW queue	Sum
Db_ActAPW	ActAPW_BufsCkpt	Decimal	Number of buffers checkpointed	Sum

Filename	Field name	Data type	Description	Compaction status
Db_ActAPW	ActAPW_BufsScanned	Decimal	Number of buffers scanned by the APWs	Sum
Db_ActAPW	ActAPW_CkptQWrites	Decimal	Number of buffer writes from the checkpoint queue	Sum
Db_ActAPW	ActAPW_Ckpts	Decimal	Number of checkpoints	Sum
Db_ActAPW	ActAPW_DBWrites	Decimal	Number of database writes performed by all APWs	Sum
Db_ActAPW	ActAPW_Flushed	Decimal	Number of buffers flushed at checkpoint	Sum
Db_ActAPW	ActAPW_Marked	Decimal	Number of buffers marked for writing	Sum
Db_ActAPW	ActAPW_ScanCycles	Decimal	Number of times an APW cycled through its tasks	Sum
Db_ActAPW	ActAPW_ScanWrites	Decimal	Number of scan buffer writes	Sum
Db_ActAPW	ActAPW_TotDBWrites	Decimal	Number of database writes performed by all processes	Sum
Db_ActAPW	ActAPW_APWEnq	Decimal	Number of buffers added to the APW queue	Sum
Db_ActAPW	ActAPW_User1	Character	User-definable field	Delete
Db_ActAPW	Sample_ID	Integer	Identification number of the gathered sample	Modify
Db_ActBuf	ActBuf_Deferred	Decimal	Number of deferred buffer writes	Sum
Db_ActBuf	ActBuf_Flushed	Decimal	Database buffers flushed at checkpoint	Sum
Db_ActBuf	ActBuf_LogicReads	Decimal	Number of logical database buffer reads performed by all processes	Sum

Filename	Field name	Data type	Description	Compaction status
Db_ActBuf	ActBuf_LogicWrites	Decimal	Number of logical database writes performed by all processes	Sum
Db_ActBuf	ActBuf_OSReads	Decimal	Number of physical OS database buffer reads performed by all processes	Sum
Db_ActBuf	ActBuf_OSWrites	Decimal	Number of physical OS database writes performed by all processes	Sum
Db_ActBuf	ActBuf_User1	Character	User-definable field	Delete
Db_ActBuf	Sample_ID	Integer	Identification number of the gathered sample	Modify
Db_ActIdx	ActIdx_Create	Decimal	Number of index keys created	Sum
Db_ActIdx	ActIdx_Delete	Decimal	Number of index keys deleted	Sum
Db_ActIdx	ActIdx_Find	Decimal	Number of index keys requested	Sum
Db_ActIdx	ActIdx_Free	Decimal	Number of index blocks placed on the free chain	Sum
Db_ActIdx	ActIdx_Remove	Decimal	Number of deleted blocks removed at the end of a transaction	Sum
Db_ActIdx	ActIdx_Splits	Decimal	Number of index block splits performed	Sum
Db_ActIdx	ActIdx_User1	Character	User-definable field	Delete
Db_ActIdx	Sample_ID	Integer	Identification number of the gathered sample	Modify
Db_ActIOFile	ActIOFile_BufReads	Decimal	Number of buffered reads from the file	Sum
Db_ActIOFile	ActIOFile_BufWrites	Decimal	Number of buffered writes to the file	Sum

Filename	Field name	Data type	Description	Compaction status
Db_ActIOFile	ActIOFile_Extends	Decimal	Number of times the file was extended	Sum
Db_ActIOFile	ActIOFile_FileName	Character	Name of the file being reported on	Delete
Db_ActIOFile	ActIOFile_Reads	Decimal	The number of times the file was read—buffered and unbuffered	Sum
Db_ActIOFile	ActIOFile_UnbufReads	Decimal	The number of times the file was read in unbuffered mode	Sum
Db_ActIOFile	ActIOFile_UnbufWrites	Decimal	The number of times the file was written in unbuffered mode	Sum
Db_ActIOFile	ActIOFile_User1	Character	User-definable field	Delete
Db_ActIOFile	ActIOFile_Writes	Decimal	The number of writes to the file—buffered and unbuffered	Sum
Db_ActIOFile	Sample_ID	Integer	Sequential identification number of the resource	Modify
Db_ActIOType	ActIOType_AiReads	Decimal	Number of AI blocks read	Sum
Db_ActIOType	ActIOType_AiWrites	Decimal	Number of AI blocks written	Sum
Db_ActIOType	ActIOType_BiReads	Decimal	Number of BI blocks read	Sum
Db_ActIOType	ActIOType_BiWrites	Decimal	Number of BI blocks written	Sum
Db_ActIOType	ActIOType_DataReads	Decimal	Number of RM blocks read	Sum
Db_ActIOType	ActIOType_DataWrites	Decimal	Number of RM blocks written	Sum
Db_ActIOType	ActIOType_IdxReads	Decimal	Number of index blocks read	Sum
Db_ActIOType	ActIOType_IdxWrites	Decimal	Number of index blocks written	Sum

Filename	Field name	Data type	Description	Compaction status
Db_ActIOType	ActIOType_User1	Character	User-definable field	Delete
Db_ActIOType	Sample_ID	Integer	Sequential identification number of the resource	Modify
Db_ActLock	ActLock_CanclReq	Decimal	Number of lock requests canceled	Sum
Db_ActLock	ActLock_Downgrade	Decimal	Number of downgraded locks	Sum
Db_ActLock	ActLock_ExclFind	Decimal	Number of times an unused, exclusive lock was down	Sum
Db_ActLock	ActLock_ExclLock	Decimal	Number of exclusive locks granted	Sum
Db_ActLock	ActLock_ExclReq	Decimal	Number of exclusive locks requested by all processes	Sum
Db_ActLock	ActLock_ExclWait	Decimal	Number of times a process had to wait for an exclusive lock	Sum
Db_ActLock	ActLock_RecGetLock	Decimal	Number of no-lock requests granted	Sum
Db_ActLock	ActLock_RecGetReq	Decimal	Number of no-lock records requested for all processes	Sum
Db_ActLock	ActLock_RecGetWait	Decimal	Number of times a process waited for a no-lock	Sum
Db_ActLock	ActLock_ShrFind	Decimal	Number of times an unused shared lock was downgraded	Sum
Db_ActLock	ActLock_ShrLock	Decimal	Number of shared locks granted	Sum
Db_ActLock	ActLock_ShrReq	Decimal	Number of shared locks requested for all processes	Sum
Db_ActLock	ActLock_ShrWait	Decimal	Number of times a process waited for a shared lock	Sum

Filename	Field name	Data type	Description	Compaction status
Db_ActLock	ActLock_UpgLock	Decimal	Number of upgrade lock requests granted	Sum
Db_ActLock	ActLock_UpgReq	Decimal	Number of lock upgrades requested for all processes	Sum
Db_ActLock	ActLock_UpgWait	Decimal	Number of times a process waited for a lock upgrade	Sum
Db_ActLock	ActLock_User1	Character	User-definable field	Delete
Db_ActLock	Sample_ID	Integer	Identification number of the gathered sample	Modify
Db_ActLog	ActLog_AIBBufWaits	Decimal	Number of AI busy buffer waits experienced	Sum
Db_ActLog	ActLog_AIByteWrites	Decimal	Number of AI bytes written to the after-image file	Sum
Db_ActLog	ActLog_AINoBufAvail	Decimal	Number of times there were no AI buffers available	Sum
Db_ActLog	ActLog_AIPartialWrites	Decimal	Number of partial AI buffer writes performed	Sum
Db_ActLog	ActLog_AIRecWrites	Decimal	Number of notes written to the AI file	Sum
Db_ActLog	ActLog_AITotWrites	Decimal	Number of AI writes performed by all processes	Sum
Db_ActLog	ActLog_AIWWrites	Decimal	Number of AI writes performed by the AIW	Sum
Db_ActLog	ActLog_BIBBufWaits	Decimal	Number of BI busy buffer waits experienced	Sum
Db_ActLog	ActLog_BIByteReads	Decimal	Number of BI bytes read by all processes	Sum
Db_ActLog	ActLog_BIByteWrites	Decimal	Number of BI bytes written to the BI file	Sum
Db_ActLog	ActLog_BIEBufWaits	Decimal	Number of BI empty buffer waits experienced	Sum

Filename	Field name	Data type	Description	Compaction status
Db_ActLog	ActLog_BIPartialWrites	Decimal	Number of partial BI buffer writes performed	Sum
Db_ActLog	ActLog_BIRecReads	Decimal	Number of BI notes read by all processes	Sum
Db_ActLog	ActLog_BIRecWrites	Decimal	Number of all BI notes written to the BI file	Sum
Db_ActLog	ActLog_BITotReads	Decimal	Number of BI block reads performed by all processes	Sum
Db_ActLog	ActLog_BITotWrites	Decimal	Number of BI writes performed by all processes	Sum
Db_ActLog	ActLog_BIWWrites	Decimal	Number of BI writes performed by the BIW	Sum
Db_ActLog	ActLog_User1	Character	User-definable field	Delete
Db_ActLog	Sample_ID	Integer	Identification number of the gathered sample	Modify
Db_ActRec	ActRec_ByteCreate	Decimal	Size, in bytes, created by all processes	Sum
Db_ActRec	ActRec_ByteDel	Decimal	Number of bytes deleted by all processes	Sum
Db_ActRec	ActRec_ByteRead	Decimal	Number of bytes read by all processes	Sum
Db_ActRec	ActRec_ByteUpd	Decimal	Number of bytes updated by all processes	Sum
Db_ActRec	ActRec_FragCreate	Decimal	Number of record fragment creates performed by all processes	Sum
Db_ActRec	ActRec_FragDel	Decimal	Number of record fragment deletes performed by all processes	Sum

Filename	Field name	Data type	Description	Compaction status
Db_ActRec	ActRec_FragRead	Decimal	Number of record fragment reads performed by all processes	Sum
Db_ActRec	ActRec_FragUpd	Decimal	Number of record fragment updates performed by all processes	Sum
Db_ActRec	ActRec_RecCreate	Decimal	Number of record creates performed by all processes	Sum
Db_ActRec	ActRec_RecDel	Decimal	Number of record deletes performed by all processes	Sum
Db_ActRec	ActRec_RecLock	Decimal	Number of record locks created for all processes	Sum
Db_ActRec	ActRec_RecRead	Decimal	Number of record reads performed by all processes	Sum
Db_ActRec	ActRec_RecUpd	Decimal	Number of record updates performed by all processes	Sum
Db_ActRec	ActRec_RecWait	Decimal	Number of record waits for all processes	Sum
Db_ActRec	ActRec_User1	Character	User-definable field	Delete
Db_ActRec	Sample_ID	Integer	Identification number of the gathered sample	Modify
Db_ActServ	ActServ_ByteRec	Decimal	Number of bytes received from the client for this server	Sum
Db_ActServ	ActServ_ByteSent	Decimal	Number of bytes sent to the client for the server	Sum
Db_ActServ	ActServ_CurrUsers	Decimal	Number of current logins to the server	Average
Db_ActServ	ActServ_Logins	Int64	Highest number of simultaneous logins for the server	Average

Filename	Field name	Data type	Description	Compaction status
Db_ActServ	ActServ_MaxUsers	Decimal	Maximum number of logins to the server	Average
Db_ActServ	ActServ_MsgRec	Decimal	Number of `promsgs` received from the client	Sum
Db_ActServ	ActServ_MsgSent	Decimal	Number of `promsgs` sent to the client for the server	Sum
Db_ActServ	ActServ_Num	Decimal	Server number for the database	Save
Db_ActServ	ActServ_PID	Integer	Process identification number of the server	Delete
Db_ActServ	ActServ_PortNum	Integer	Port number on which the server operates	Delete
Db_ActServ	ActServ_QryRec	Decimal	Number of queries received from the client for the server	Sum
Db_ActServ	ActServ_RecRec	Decimal	Number of records received from the client for the server	Sum
Db_ActServ	ActServ_RecSent	Decimal	Number of records sent to the client for the server	Sum
Db_ActServ	ActServ_TimeSlice	Decimal	Number of times the limit number of records for a request has been interrupted to service another	Sum
Db_ActServ	ActServ_Type	Character	The type of server (SQL, ABL, etc.)	Save
Db_ActServ	ActServ_User1	Character	User-definable field	Delete
Db_ActServ	Sample_ID	Integer	Identification number of the gathered sample	Modify
Db_ActSum	ActSum_AllocNewRm	Decimal	Number of times space was allocated on the RM	Sum
Db_ActSum	ActSum_BytesAlloc	Decimal	Number of bytes allocated for records	Sum

Filename	Field name	Data type	Description	Compaction status
Db_ActSum	ActSum_Examined	Decimal	Number of examined data blocks on the RM chain	Sum
Db_ActSum	ActSum_FromFree	Decimal	Number of times space was allocated from the free chain	Sum
Db_ActSum	ActSum_FromRm	Decimal	Number of times space was allocated from the RM chain	Sum
Db_ActSum	ActSum_Removed	Decimal	Number of blocks removed from the RM chain	Sum
Db_ActSum	ActSum_RetFree	Decimal	Number of times a free block was returned to the free chain	Sum
Db_ActSum	ActSum_TakeFree	Decimal	Number of times a free block was taken from the free chain	Sum
Db_ActSum	ActSum_AiWrites	Decimal	Number of AI block writes performed by all processes	Sum
Db_ActSum	ActSum_BiReads	Decimal	Number of BI block reads performed by all processes	Sum
Db_ActSum	ActSum_BiWrites	Decimal	Number of BI block writes performed by all processes	Sum
Db_ActSum	ActSum_Commits	Decimal	Number of commits performed by all processes	Sum
Db_ActSum	ActSum_DbAccesses	Decimal	Number of database block requests by all processes	Sum
Db_ActSum	ActSum_DbExtend	Decimal	Number of times the database was extended	Sum
Db_ActSum	ActSum_DbReads	Decimal	Number of database block reads performed by all processes	Sum

Filename	Field name	Data type	Description	Compaction status
Db_ActSum	ActSum_DbWrites	Decimal	Number of database block writes performed by all processes	Sum
Db_ActSum	ActSum_Undos	Decimal	Number of transaction undos or backouts performed	Sum
Db_ActSum	ActSum_User1	Character	User-definable field	Delete
Db_ActSum	Sample_ID	Integer	Identification number of the gathered sample	Modify
Db_AreaStatus	Sample_ID	Integer	Identification number of the gathered sample	Modify
Db_AreaStatus	AreaStatus_Areaname	Character	Name of the area	Save
Db_AreaStatus	AreaStatus_Areanum	Decimal	Number of the area	Save
Db_AreaStatus	AreaStatus_TotBlocks	Decimal	Total number of blocks in the area	Average
Db_AreaStatus	AreaStatus_Hiwater	Decimal	Largest number of blocks to which the database has grown	Average
Db_AreaStatus	AreaStatus_Extents	Decimal	Number of extents in the area	Save
Db_AreaStatus	AreaStatus_FreeNum	Int64	Number of free blocks	Average
Db_AreaStatus	AreaStatus_Rmnum	Int64	Number of Record Manager (RM) blocks on the RM chain	Average
Db_AreaStatus	AreaStatus_User1	Character	User-definable field	Delete
Db_Checkpoint	Checkpoint_ApwQ	Decimal	Number of buffers placed in the APW write queue	Average
Db_Checkpoint	Checkpoint_CkptQ	Decimal	Number of buffers placed in the checkpoint queue	Average
Db_Checkpoint	Checkpoint_Date	Date	The date the checkpoint started	Delete

Filename	Field name	Data type	Description	Compaction status
Db_Checkpoint	Checkpoint_Dirty	Decimal	Number of dirty buffers at the beginning of the checkpoint	Average
Db_Checkpoint	Checkpoint_Flush	Decimal	Number of buffers flushed at the end of the checkpoint	Average
Db_Checkpoint	Checkpoint_Length	Character	Length, in seconds, of the checkpoint	Average
Db_Checkpoint	Checkpoint_Num	Decimal	Checkpoint number	Average
Db_Checkpoint	Checkpoint_Scan	Decimal	Number of buffers scanned by all of the APWs	Average
Db_Checkpoint	Checkpoint_Time	Character	Begin time for the checkpoint	Average
Db_Checkpoint	Checkpoint_User1	Character	User-definable field	Delete
Db_Checkpoint	Sample_ID	Integer	Identification number of the sample	Modify
Db_IndexStat	IndexStat_Blockdelete	Decimal	Number of blocks deleted from the index by all processes	Sum
Db_IndexStat	IndexStat_Create	Decimal	Number of keys created in the index by all processes	Sum
Db_IndexStat	IndexStat_Delete	Decimal	Index ID in the database	Sum
Db_IndexStat	IndexStat_IndexName	Character	The name of the index for this information	Save
Db_IndexStat	IndexStat_IdxNum	Int64	Index number within the database	Save
Db_IndexStat	IndexStat_Read	Decimal	Number of keys read in the index by all processes	Sum
Db_IndexStat	IndexStat_Split	Decimal	Number of blocks split in the index by all processes	Sum
Db_IndexStat	IndexStat_User1	Character	User-definable field	Delete

Filename	Field name	Data type	Description	Compaction status
Db_IndexStat	Sample_ID	Integer	Identification number of the gathered sample	Modify
Db_IndexStat	IndexStat_TableName	Character	The name of the table for this information	Save
Db_TableStat	Sample_ID	Integer	Identification number of the gathered sample	Modify
Db_TableStat	TableStat_Create	Decimal	Number of records created in the table by all processes	Sum
Db_TableStat	TableStat_Delete	Decimal	Number of records deleted from the table by all processes	Sum
Db_TableStat	TableStat_Read	Decimal	Number of records read in the table by all processes	Sum
Db_TableStat	TableStat_Table_ID	Decimal	Table number within the database	Save
Db_TableStat	TableStat_TableName	Character	The name of the table for this information	Save
Db_TableStat	TableStat_Update	Decimal	Number of records updated in the table by all processes	Sum
Db_TableStat	TableStat_User1	Character	User-definable field	Delete

Compaction details for database tables

The following table summarizes how the compaction process affects data rows contained in the database tables.

Table 8: Configuration tables and the compaction process

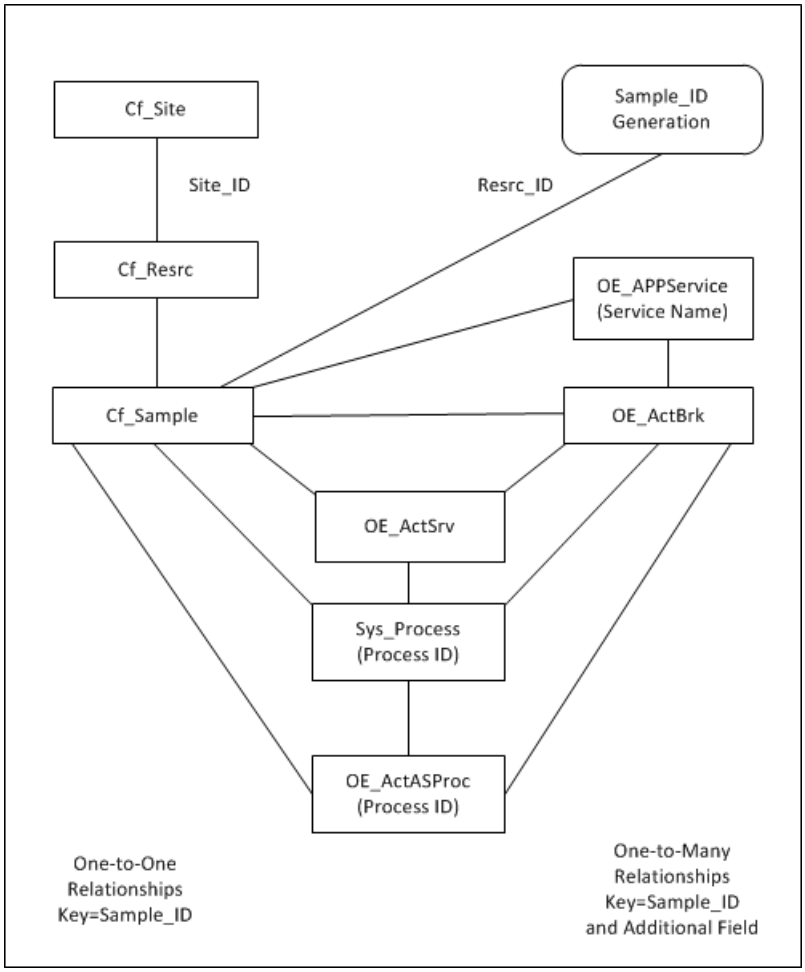
Database table	What happens to table data during compaction
Db_ActAPW	This is a multi-row table in which most rows are summed during the compaction process.
Db_ActBuf	This is a multi-row table in which most rows are summed during the compaction process.

Database table	What happens to table data during compaction
Db_ActIdx	This is a multi-row table in which most rows are summed during the compaction process.
Db_ActIOFile	This is a multi-row table. The key to this table is Sample_ID and FileName.
Db_ActLock	This is a multi-row table in which most rows are summed during the compaction process.
Db_ActRec	This is a multi-row table in which most rows are summed during the compaction process.
Db_ActSum	This is a multi-row table in which most rows are summed during the compaction process.
Db_Checkpoint	Data is deleted and not compacted.
Db_TableStat	This is a multi-row table. The key to this table is Sample_ID and TableID.
Db_IndexStat	This is a multi-row table. The key to this table is Sample_ID and IndexNum.
Db_ActServ	The compaction process reflects a mixing of sums and averages. This is a multi-row table. The key to this table is Sample_ID and Server Number. Every time you start and stop the database, the servers change.
DB_ActLog	This is a multi-row table in which most rows are summed during the compaction process.
DB_ActIOType	This is a multi-row table in which most rows are summed during the compaction process.
Db_AreaStatus	This is a multi-row table. The key to this table is Sample_ID and Area Name.

OpenEdge tables

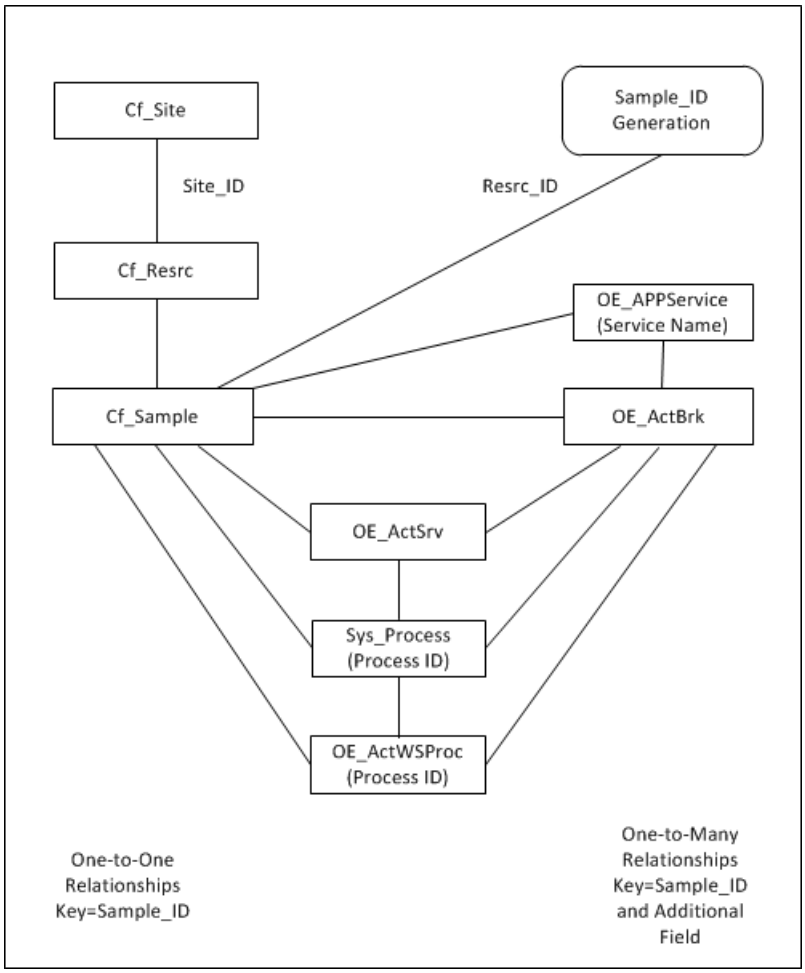
The schema of OpenEdge tables is dependent upon many subsystems. The following table shows the schema of AppServer tables.

Figure 3: OpenEdge AppServer table relationships



The following table shows the schema of WebSpeed tables.

Figure 4: OpenEdge WebSpeed table relationships



If you configure OpenEdge Management to monitor WebSpeed and/or AppServer resources, OpenEdge Management stores the data in the tables listed in the following table.

Table 9: OpenEdge tables

Table name	Table contents	Description
ActASProc	AppServer Data	Contains information about each procedure executed for the AppServer
ActBrk	Broker Data	Contains information about AppServer and WebSpeed broker activity requests
ActSrv	Server Data	Contains information about OpenEdge broker server activity
ActWSProc	WebSpeed Data	Contains information about each procedure executed for WebSpeed
APPService	Application Data	Contains information about application services

The following table describes each OpenEdge table including fields, data types, brief descriptions, and how data for each field is handled during data compaction. For more information about the filename prefix, see the Table prefix definitions table.

Table 10: OpenEdge table details

Filename	Field name	Data type	Description	Compaction status
OE_ActASProc	Sample_ID	Integer	Identification number of the gathered sample	Modify
OE_ActASProc	ActBrk_PID	Decimal	Identification number of the broker process	Save
OE_ActASProc	ActBrk_UUID	Character	Broker UUID for the broker process	Save
OE_ActASProc	ActASProc_Name	Character	Name of the procedure	Save
OE_ActASProc	ActASProc_Calls	Decimal	Number of times the procedure has been called	Sum
OE_ActASProc	ActASProc_TotTime	Decimal	Cumulative time the procedure has run	Sum
OE_ActASProc	ActASProc_MaxTime	Decimal	Longest amount of time that the procedure has run	Average
OE_ActASProc	ActASProc_Parent	Character	Name of the parent procedure	Save
OE_ActASProc	ActASProc_ProcType	Character	Type of procedure: internal, persistent, or user-defined	Save
OE_ActASProc	ActASProc_SuccessCount	Decimal	Number of times the procedure completed with a success condition	Sum
OE_ActASProc	ActASProc_ErrorCount	Decimal	Number of times the procedure completed with an error condition	Sum
OE_ActASProc	ActASProc_QuitCount	Decimal	Number of times the procedure completed with a quit condition	Sum
OE_ActASProc	ActASProc_StopCount	Decimal	Number of times the procedure completed with a stop condition	Sum

Filename	Field name	Data type	Description	Compaction status
OE_ActASProc	ActASProc_DebugCount	Decimal	Number of times the procedure completed with a debug condition	Sum
OE_ActBrk	Sample_ID	Integer	Identification number of the gathered sample	Modify
OE_ActBrk	ActBrk_PID	Decimal	Process identification number of the broker process	Save
OE_ActBrk	ActBrk_UUID	Character	UUID of the broker process	Save
OE_ActBrk	ActBrk_Name	Character	Name of the broker	Save
OE_ActBrk	ActBrk_ReqRec	Decimal	Number of requests received by the broker	Sum
OE_ActBrk	ActBrk_ReqComp	Decimal	Number of requests completed by the broker	Sum
OE_ActBrk	ActBrk_ReqQueued	Decimal	Number of broker requests queued	Sum
OE_ActBrk	ActBrk_ReqRejected	Decimal	Number of requests rejected by the broker	Sum
OE_ActBrk	ActBrk_CurrClients	Decimal	Current client count for the broker	Average
OE_ActBrk	ActBrk_TotClients	Decimal	Total number of clients that have connected to the broker	Sum
OE_ActBrk	ActBrk_CurrServers	Integer	Current number of servers	Average
OE_ActBrk	ActBrk_Mode	Character	Mode of the broker and servers: stateless, state-aware, state-reset, or state free	Save
OE_ActBrk	ActBrk_CurrQueue	Integer	Current number of queued requests	Average
OE_ActSrv	Sample_ID	Integer	Identification number of the gathered sample	Modify

Filename	Field name	Data type	Description	Compaction status
OE_ActSrv	ActBrk_PID	Decimal	Process identification number of the broker process	Save
OE_ActSrv	ActBrk_UUID	Character	UUID of the broker process	Save
OE_ActSrv	ActSrv_PID	Decimal	Process identification number of the server process	Save
OE_ActSrv	ActSrv_Status	Character	Current status of the server	Save
OE_ActSrv	ActSrv_IdleCount	Decimal	Number of times an agent has become idle	Sum
OE_ActSrv	ActSrv_IdleTime	Decimal	Amount of time an agent has spent idle	Sum
OE_ActSrv	ActSrv_BusyCount	Decimal	Number of times an agent has had a status of busy	Sum
OE_ActSrv	ActSrv_BusyTime	Decimal	Amount of time an agent has had a busy status	Sum
OE_ActSrv	ActSrv_LockedCount	Decimal	Number of times an agent has had a status of locked	Sum
OE_ActSrv	ActSrv_LockedTime	Decimal	Amount of time an agent has had a status of locked	Sum
OE_ActWSProc	Sample_ID	Integer	Identification number of the gathered sample	Modify
OE_ActWSProc	ActBrk_PID	Decimal	Process identification number of the broker process	Save
OE_ActWSProc	ActBrk_UUID	Character	UUID of the broker process	Save
OE_ActWSProc	ActWSProc_Name	Character	Name of the procedure	Save
OE_ActWSProc	ActWSProc_Calls	Decimal	Number of times the procedure has been called	Sum

Filename	Field name	Data type	Description	Compaction status
OE_ActWSProc	ActWSProc_TotTime	Decimal	Amount of time the process has run	Sum
OE_ActWSProc	ActWSProc_MaxTime	Decimal	Longest amount of time the procedure has run	Average
OE_ActWSProc	ActWSProc_Parent	Character	Parent procedure	Save
OE_ActWSProc	ActWSProc_ProcType	Character	Procedure type: internal, persistent, or user-defined	Save
OE_APPService	Sample_ID	Integer	Identification number of the gathered sample	Modify
OE_APPService	APPS_ServiceName	Character	Application service name	Save
OE_APPService	ActBrk_UUID	Character	UUID of the broker process	Save

System tables

If you configure OpenEdge Management to monitor system resources, it stores the gathered data in the Trend Database system tables. The following table lists each system table's contents.

Table 11: System tables

Table name	Table contents	Description
Sys_CPU	CPU data	Contains CPU data for OpenEdge Management instances that are trended to the OpenEdge Management Trend Database.
Sys_Mem	Memory data	Contains Memory data for OpenEdge Management instances that are trended to the OpenEdge Management Trend Database.
Sys_Dev	Disk Device data	Contains Disk Device data for OpenEdge Management instances that are trended to the OpenEdge Management Trend Database. The disk device is typically a local drive, but can also be an NFS-type drive.
Sys_FileSys	File System data	Contains File System data for OpenEdge Management instances that are trended to the OpenEdge Management Trend Database.

Table name	Table contents	Description
Sys_FileSize	File Size data	Contains file size data for monitored files.
Sys_Process	Process data	Contains data about OpenEdge server processes.

The following table lists the field names, the data types, a brief description of each, and how data for that field is handled when data compaction activities occur for each system table. For more information about the filename prefix, see the Table prefix definitions table.

Table 12: System table details

Filename	Field name	Data type	Description	Compaction status
Sys_CPU	CPU_System	Decimal	Percentage of system activity on the CPU	Average
Sys_CPU	CPU_Idle	Integer	Percentage of time that the CPU is idle	Average
Sys_CPU	CPU_User1	Character	User-definable field	Delete
Sys_CPU	CPU_Usr	Decimal	Percentage of user activity on the CPU	Average
Sys_CPU	CPU_Wait	Decimal	Percentage of wait I/O on the CPU	Average
Sys_CPU	Sample_ID	Integer	Identification number of the gathered sample	Modify
Sys_Dev	Dev_AvgQue	Decimal	Average of queue activity on the disk	Sum
Sys_Dev	Dev_AvgServ	Decimal	Average of service time on the disk	Sum
Sys_Dev	Dev_AvgWait	Decimal	Average wait time on the disk	Sum
Sys_Dev	Dev_Name	Character	Name of the device	Save
Sys_Dev	Dev_Pct_Busy	Decimal	Percentage of time the device was busy	Average
Sys_Dev	Dev_User1	Character	User-identifiable field	Delete
Sys_Dev	Sample_ID	Integer	Identification number of the gathered sample	Modify
Sys_Mem	Mem_PgIn	Integer	Number of memory pages paged in	Sum

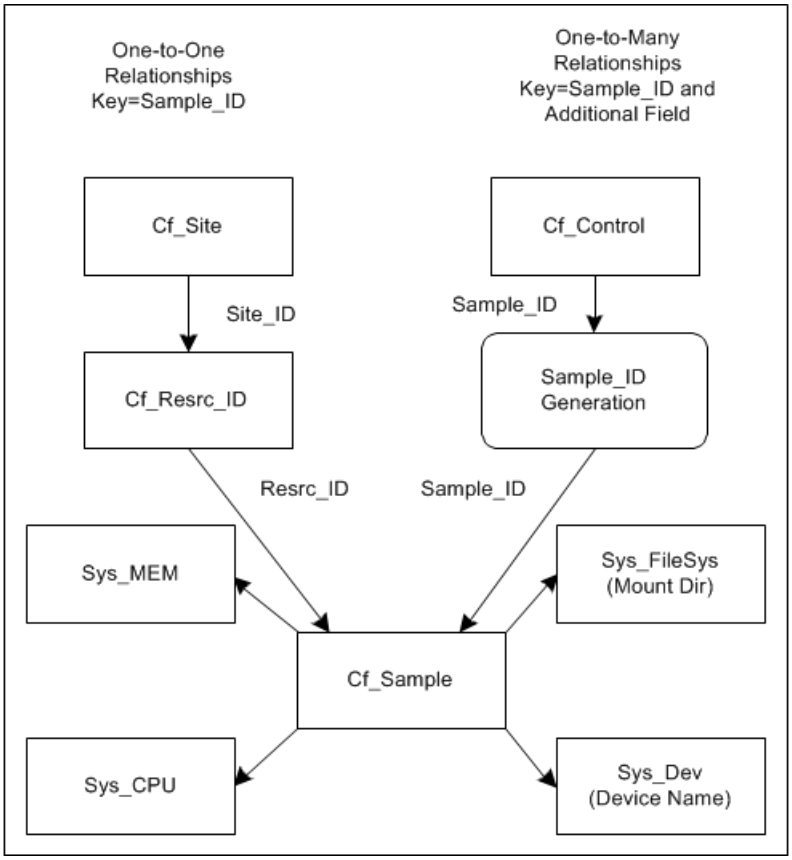
Filename	Field name	Data type	Description	Compaction status
Sys_Mem	Mem_PgOut	Integer	Number of memory pages paged out	Sum
Sys_Mem	Mem_SysFree	Decimal	Physical memory in system blocks free on the system	Average
Sys_Mem	Mem_SysTot	Decimal	Physical memory in system blocks available on the machine	Average
Sys_Mem	Mem_User1	Character	User-definable field	Delete
Sys_Mem	Mem_VirtFree	Decimal	Virtual memory in system blocks free on the system	Sum
Sys_Mem	Mem_VirtTot	Decimal	Virtual memory in system blocks available on the machine	Sum
Sys_Mem	Sample_ID	Integer	Identification number of the gathered sample	Modify
Sys_FileSize	FileSize_Name	Character	Name of the file system	Save
Sys_FileSize	FileSize_Size	Decimal	File size in bytes	Average
Sys_FileSize	FileSize_User1	Character	User-definable field	Delete
Sys_FileSize	Sample_ID	Integer	Identification number of the gathered sample	Modify
Sys_FileSys	FileSys_Avail	Decimal	Number of available blocks on the file system	Average
Sys_FileSys	FileSys_Capacity	Integer	Number of total blocks on the file system	Average
Sys_FileSys	FileSys_Mount	Character	Name of directory where the file system is mounted	Save
Sys_FileSys	FileSys_Name	Character	Name of the file system	Save
Sys_FileSys	FileSys_UsedPct	Decimal	Percentage of the file system used	Average
Sys_FileSys	FileSys_User1	Character	User-definable field	Delete
Sys_FileSys	Sample_ID	Integer	Identification number of the gathered sample	Modify

Filename	Field name	Data type	Description	Compaction status
Sys_Process	Process_PID	Decimal	Process ID of the process being monitored	Save
Sys_Process	Process_PPID	Decimal	Parent process ID of the process being monitored	Save
Sys_Process	Process_CPUUser	Decimal	Amount of CPU time (in seconds) dedicated to user	Sum
Sys_Process	Process_CPUSystem	Decimal	Amount of CPU time (in seconds) dedicated to system	Sum
Sys_Process	Process_ElapsedTime	Decimal	Elapsed process time (in seconds) since last poll	Average
Sys_Process	Process_CPUPct	Decimal	Percentage of CPU usage since last poll	Average
Sys_Process	Process_CPUPctWT	Decimal	Percentage of CPU usage since last poll (weighted per CPU)	Average
Sys_Process	Process_UserName	Character	Name of the user attached to the process	Save
Sys_Process	Process_UserGroup	Character	Name of the group related to the process	Save
Sys_Process	Process_Command	Character	The command executed. **Note:** This field contains only the executable.	Save
Sys_Process	Process_CommandArgs	Character	Arguments fed to the command	Save
Sys_Process	Process_VirtMem	Decimal	Virtual memory size (as defined by the operating system)	Sum
Sys_Process	Process_ResMemory	Decimal	Resident memory size (as defined by the operating system)	Sum

Filename	Field name	Data type	Description	Compaction status
Sys_Process	Process_StartTimeStamp	Character	Date and time that the process started	Save
Sys_Process	Sample_ID	Integer	Identification number of the gathered sample	Modify

The schema of system tables is dependent on many subsystems. The following figure illustrates the relationship among the different system tables.

Figure 5: System table relationships



The above figure describes the data gathered about trended system information. Every Sys_ table contains the sample's ID number. The sample table stores data from the gathered sample that applies to all the tables. The sample table also contains the sample's Site and Resource IDs. A control record in the control table generates the Sample ID.

Compaction details for system tables

The following table summarizes how the compaction process affects data rows contained in the specified tables.

Table 13: System tables and the compaction process

System table	What happens to table data during compaction
Sys_CPU	The averages will be stored.
Sys_Dev	This is a multi-row table. The key to this table is Sample_ID and Device Name.
Sys_Mem	The content reflects a mixing of compaction and averages.
Sys_FileSize	This is a multi-row table in which averages are stored. The key to this table is Sample_ID and Name 2 additional fields—Max and Min.
Sys_FileSys	This is a multi-row table in which averages are stored. The key to this table is Sample_ID and Mount.
Sys_Process	This is a multi-row table. The key to this table is Sample_ID and Process PID.

Archive tables

You can archive data associated with database, OpenEdge, and system tables. You cannot archive data stored in configuration tables.

This section identifies:

- The database, OpenEdge, and system tables for which you can archive data. The following table identifies the database tables that you can archive, the *Archive system tables* table identifies the system tables that you can archive, and the *Archive OpenEdge tables* table identifies the OpenEdge tables that you can archive.
- Common archive-related fields that are generated for each table you archive.
- An example of archive data generated for an archived table.

The following table identifies and briefly describes each OpenEdge Management Trend Database table that you can archive.

Table 14: Archive OpenEdge Management Trend Database tables

Table name	Table contents	Description
Ar_ActAPW	Information from the _ActAPW VST	Archives APW activity data, such as the number of buffer reads and writes
Ar_ActBuf	Information from the _ActBuffer VST	Archives the summary of buffer activity in the first sample; the primary buffer activity in the second sample; and the alternate buffer activity in the third sample
Ar_ActIdx	Information from the _Index and _ActIndex VSTs	Archives index data about overall performance
Ar_ActIOFile	Information from the _ActIOFile VST	Archives I/O data about each database file, except for the `.lk` and `.lg` files

Table name	Table contents	Description
Ar_ActIOType	Information from the _ActIOType VST	Archives database I/O data, such as the number of BI and AI blocks read
Ar_ActLock	Information from the _ActLock and _LockReq VSTs	Archives lock activity data, such as the number of requests and the number of database locks
Ar_ActLog	Information from the _ActAILog and _ActBILog VSTs	Archives activity data about before- and after-imaging
Ar_ActRec	Information from the _ActRecord and _ActSummary VSTs	Archives performance data about record reads, writes, and updates
Ar_ActServ	Information from the _ActServer and _Server VSTs	Archives server process data, such as bytes and records sent and received and the number of users
Ar_AreaStatus	Information from the _AreaStatus VST	Archives data about each area's HWM, Free Chain, and RM Chain
Ar_ActSum	Information from the _ActSummary VST	Archives database activity data, such as the number of records created and deleted
Ar_Checkpoint	Information from the _Checkpoint VST	Archives data about checkpoint operations
Ar_IndexStat	Information from the _IndexStat and _Index VSTs	Archives individual index data about reads and writes

The following table identifies and briefly describes each system table that you can archive.

Table 15: Archive system tables

Table name	Table contents	Description
Ar_CPU	CPU data	Archives CPU data for the host system.
Ar_Mem	Memory data	Archives Memory data for the host system.
Ar_Dev	Disk Device data	Archives Disk Device data for the host system. The disk device is typically a local drive, but can also be an NFS-type drive.
Ar_FileSys	File System data	Archives File System data for the host system.
Ar_FileSize	File Size data	Archives File Size data for monitored files.

The following table identifies and briefly describes each OpenEdge table that you can archive.

Table 16: Archive OpenEdge tables

Table name	Table contents	Description
Ar_OE_ActASProc	AppServer data	Archives information about each procedure executed for the AppServer
Ar_OE_ActBrk	Broker data	Archives information about AppServer and WebSpeed broker activity requests
Ar_OE_ActServ	Server data	Archives information about AppServer and WebSpeed server and agent activity
Ar_OE_ActWSProc	WebSpeed data	Archives information about each procedure executed for WebSpeed
Ar_OE_AppService	Application data	Archives information about application services

Common archive fields

You can archive data that is derived from the values stored in fields associated with database, OpenEdge, or system tables. An additional set of archive-specific fields for which unique values are defined during the archiving process is also generated for, and stored with, each archived table.

The *Archive OpenEdge Management Trend Database tables* and *Archive system tables* tables define the unique fields associated with each database or system table. The following table identifies the common fields for which values are generated and added to a table when you archive it.

Table 17: Common archive fields

Field name	Data type	Description
_ArchiveType	Character	Identifies the field's archive type: • **H** — Hourly • **D** — Daily • **W** — Weekly
_Length	Integer	Number of seconds for the archive to take place
_StartDate	Date	Start date for the row (this date will always be the same as the end date for hourly and daily)
_StartTime	Integer	Start time for the row
_EndDate	Date	End date for the row
_EndTime	Integer	End time for the row
_Transactions	Integer	Number of transactions committed
_RecordCount	Integer	Number of records in this archived file

Field name	Data type	Description
Resrc_ID	Integer	The Resource ID of the database for this site
Site_ID	Character	The Site ID for the database

Complete contents of an archived table

The following table identifies the table contents of the Ar_ActIdx table (the archived version of the database table ActIdx), which stores index data about overall performance.

Table 18: Contents of the Ar_ActIdx table

Field name	Data type	Description
ActIdx_Create	Decimal	Number of index keys created
ActIdx_Delete	Decimal	Number of index keys deleted
ActIdx_Find	Decimal	Number of index keys requested
ActIdx_Free	Decimal	Number of index blocks placed on the free chain
ActIdx_Remove	Decimal	Number of deleted blocks removed at the end of a transaction
ActIdx_Splits	Decimal	Number of index block splits performed
ActIdx_User1	Character	User-definable field
Sample_ID	Integer	Identification number of the gathered sample
ActIdx_ArchiveType	Character	Identifies the field's archive type: • **H** — Hourly • **D** — Daily • **W** — Weekly
ActIdx_Length	Integer	Number of seconds for the archive to take place
ActIdx_StartDate	Date	Start date for the row (this date will always be the same as the end date for hourly and daily)
ActIdx_StartTime	Integer	Start time for the row
ActIdx_EndDate	Date	End date for the row
ActIdx_EndTime	Integer	End time for the row
ActIdx_Transactions	Integer	Number of transactions committed

Field name	Data type	Description
ActIdx_RecordCount	Integer	Number of records in this archived file
Resrc_ID	Integer	The Resource ID of the database for this site
Site_ID	Character	The Site_ID for the database

Database analysis tables

This section describes the tables associated with performing database analysis to determine trends in your data for planning and performance turning. the Db_TabAnalysis table details table shows the contents of Db_TabAnalysis, and the Db_IdxAnalysis table details table shows the contents of Db_IdxAnalysis.

Db_TabAnalysis table

The Db_TabAnalysis table stores all of the information related to table analysis for a specific table, refers to the Cf_Resrc and Cf_Site table using appropriate IDs, and stores the resource name.

Table 19: Db_TabAnalysis table details

Field name	Data type	Description	Default value
Tab_ID	Integer	The sequential ID of the row	Next ID
Tab_TableName	Character	The table name used by the table; Tab_TableName is the same as _File-name	N/A
Tab_RecordCount	Int64	The number of records in the table	N/A
Tab_Bytes	Decimal	The size, in total bytes, of all of the records in this table	N/A
Tab_MinRecSize	Integer	The size, in bytes, of the smallest record in this table	N/A
Tab_MaxRecSize	Integer	The size, in bytes, of the largest record in this table	N/A
Tab_AvgRecSize	Integer	The average record size, in bytes, for records in this table	N/A
Tab_FragmentCount	Integer	The count of how many records within the database that were required to be physically split	N/A

Field name	Data type	Description	Default value
Tab_FragmentFactor	Decimal	The factor of how many records within the database that were required to be physically split	1
Tab_ScatterFactor	Decimal	Indicates how far between records within the database	N/A
Tab_ResrcName	Character	The name of the database resource	N/A
Tab_Date	Date	The date on which the analysis was run; it is extracted from the report	N/A
Tab_Time	Integer	The time at which the analysis was run; it is extracted from the report	N/A
Site_ID[1]	Character	The Site_ID for the database	N/A
Resrc_ID[1]	Integer	The Resource ID of the database for this site	N/A
Tab_AreaName	Character	Database Table Area Name	N/A

Db_IdxAnalysis table

The Db_IdxAnalysis table stores all of the information related to index analysis for a specific table, refers back to the Cf_Resrc and Cf_Site table using appropriate IDs, and stores the resource name.

Table 20: Db_IdxAnalysis table details

Field name	Data type	Description	Default
Tab_ID	Integer	The sequential ID of the row	Next ID
IDX_TableName	Character	The table name that the index references. IDX_TableName is the same as _File-name	N/A
IDX_AreaName	Character	Database Index Area Name	N/A
IDX_Number	Int64	The index number for the index within the database	N/A
IDX_Bytes	Decimal	The size, in total bytes, of all of the index entries in this index for this table	N/A
IDX_Blocks	Decimal	The number of blocks in this index	N/A

[1] The Site_ID and Resrc_ID fields are included in the table definition. They are not required fields. However, they add value by providing the capability to tie the dbanalysis information back to the other information for that same database.

Field name	Data type	Description	Default
IDX_UtilizationPct	Decimal	The percentage of index blocks that are filled	N/A
IDX_Utilization Factor	Decimal	The average number of blocks between which the index is split	N/A
IDX_IndexName	Character	The name of the index	N/A
IDX_Levels	Decimal	Indicates how many levels deep the index traverses	1
IDX_ResrcName	Character	The name of the database resource	N/A
IDX_Date	Date	The date on which the report was run; it is extracted from the report	Report's content
IDX_Time	Integer	The time at which the report was run; it is extracted from the report	Report's content
Site_ID	Character	The Site_ID for the database	N/A
Resrc_ID	Integer	The Resource ID of the database for this site	N/A

Network table

The OpenEdge Management Trend Database stores network data in the Net_Data table. The following table shows the content of the Net_Data table.

Table 21: Net_Data table details

Field name	Data type	Description
Sample_ID	Integer	The sequential identification number of the gathered sample
Net_Time	Integer	The amount of time it took the network resource to respond to the poll
Net_Status	Character	The status of the returned poll: passed, failure, tardy, time-out, or unreachable
Net_User1	Character	User-definable field

Index

A

Archived data
- common fields *71*
- example *72*

D

Data compaction
- as a job *28–29*
- configuration tables *43*
- database tables *57*
- statistics *33*
- system tables *68*

Data Compaction jobs
- advanced information *31*
- creating *29*

Database Maintenance job templates
- advanced information *31*
- environment variables *33*
- trend data compaction *29*

L

Log file viewers
- dead *33*

Log file viewers *(continued)*
- stale *33*

O

OpenEdge Management Trend Database
- data
  - compacting *27*, *29*
  - data *29*
  - purging *28–29*
  - retaining *28*
- managing *25*
- size *26*

OpenEdge Management Trend Database schema
- analysis tables *73*
- archive tables *69*
- configuration tables *38*
- definitions *38*
- network table *75*
- OpenEdge
  - AppServer *59*
  - WebSpeed *59*
- OpenEdge tables *59*
- system tables *64*
- table and field name definitions *38*
- Trend Database tables *43*