
Administering Platform™ Analytics™

Version 7.5
April 2009

Comments to: doc@platform.com
Support: support@platform.com



Copyright

© 1994-2009, Platform Computing Inc.

Although the information in this document has been carefully reviewed, Platform Computing Inc. (“Platform”) does not warrant it to be free of errors or omissions. Platform reserves the right to make corrections, updates, revisions or changes to the information in this document.

UNLESS OTHERWISE EXPRESSLY STATED BY PLATFORM, THE PROGRAM DESCRIBED IN THIS DOCUMENT IS PROVIDED “AS IS” AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL PLATFORM COMPUTING BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION ANY LOST PROFITS, DATA, OR SAVINGS, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS PROGRAM.

We’d like to hear from you

You can help us make this document better by telling us what you think of the content, organization, and usefulness of the information. If you find an error, or just want to make a suggestion for improving this document, please address your comments to doc@platform.com.

Your comments should pertain only to Platform documentation. For product support, contact support@platform.com.

Document redistribution and translation

This document is protected by copyright and you may not redistribute or translate it into another language, in part or in whole.

Internal redistribution

You may only redistribute this document internally within your organization (for example, on an intranet) provided that you continue to check the Platform Web site for updates and update your version of the documentation. You may not make it available to your organization over the Internet.

Trademarks

LSF is a registered trademark of Platform Computing Inc. in the United States and in other jurisdictions.

ACCELERATING INTELLIGENCE, PLATFORM COMPUTING, PLATFORM SYMPHONY, PLATFORM JOBSCHEDULER, PLATFORM ENTERPRISE GRID ORCHESTRATOR, PLATFORM EGO, and the PLATFORM and PLATFORM LSF logos are trademarks of Platform Computing Inc. in the United States and in other jurisdictions.

UNIX is a registered trademark of The Open Group in the United States and in other jurisdictions.

Microsoft is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Other products or services mentioned in this document are identified by the trademarks or service marks of their respective owners.

Third-party license agreements

<http://www.platform.com/Company/third.part.license.htm>

Third-party copyright notices

<http://www.platform.com/Company/Third.Party.Copyright.htm>

Contents

Part I: About Analytics

| | | |
|---|--|----|
| 1 | Introduction to Analytics | 9 |
| 2 | Architecture overview | 11 |
| | PERF directories in the Analytics node | 11 |
| | System architecture | 13 |
| 3 | Licensing | 15 |

Part II: Database host

| | | |
|---|---------------------|----|
| 4 | Database | 19 |
| | About the database | 19 |
| 5 | Data sources | 21 |
| | Contents | 21 |
| | About data sources | 21 |
| | Data source actions | 22 |

Part III: Analytics node

| | | |
|---|--|----|
| 6 | Loader controller | 27 |
| | Contents | 27 |
| | About the loader controller | 27 |
| | Configuration to modify loader controller behavior | 28 |
| | Loader controller actions | 29 |
| 7 | Data loaders | 31 |
| | Contents | 31 |
| | About data loaders | 31 |
| | Configuration to modify data loader behavior | 33 |
| | Data loader actions | 34 |
| 8 | Analytics node command-line tools | 37 |
| | Contents | 37 |
| | dbconfig | 39 |
| | Synopsis | 39 |

| | |
|--|----|
| Description | 39 |
| Options | 39 |
| perfadmin | 41 |
| Synopsis | 41 |
| Description | 41 |
| Options | 41 |
| Output | 41 |
| plclient | 43 |
| Synopsis | 43 |
| Description | 43 |
| Options | 43 |
| 9 Analytics node configuration files | 45 |
| Contents | 45 |
| perf.conf | 47 |
| About perf.conf | 47 |
| Changing perf.conf configuration | 47 |
| Location | 47 |
| Format | 47 |
| DLP_ENABLED | 47 |
| EGO_VERSION | 48 |
| LICENSE_FILE | 48 |
| LICENSE_VERSION | 49 |
| LOADER_BATCH_SIZE | 49 |
| LSF_ENVDIR | 50 |
| LSF_VERSION | 50 |
| PERF_CONFDIR | 50 |
| PERF_LOGDIR | 50 |
| PERF_TOP | 51 |
| PERF_VERSION | 51 |
| PERF_WORKDIR | 51 |
| 10 Troubleshooting the node | 53 |
| Contents | 53 |
| Change the default log level of your log files | 53 |
| Disable data collection for individual data loaders | 54 |
| Check the status of the loader controller | 55 |
| Check the status of the data loaders | 55 |
| Check the status of the Analytics node database connection | 56 |

Part IV: Analytics server

| | |
|-----------------------------|----|
| 11 Analytics Console | 59 |
| Contents | 59 |
| About the Analytics Console | 59 |
| Analytics Console actions | 60 |
| 12 Data transformers | 61 |

| | | |
|----|---|----|
| | Contents | 61 |
| | About data transformers | 61 |
| | Configuration to modify data transformer behavior | 64 |
| | Data transformer actions | 64 |
| 13 | Events | 65 |
| | Contents | 65 |
| | About events | 65 |
| | Event actions | 65 |
| 14 | Data purger | 67 |
| | Contents | 67 |
| | About the data purger | 67 |
| | Data purger actions | 68 |
| 15 | Datamarts | 69 |
| | Contents | 69 |
| | About datamarts | 69 |
| | Datamart actions | 72 |
| 16 | Cubes | 73 |
| | Contents | 73 |
| | About cubes | 73 |
| | Cube actions | 76 |
| 17 | Scheduled tasks | 77 |
| | Contents | 77 |
| | About scheduled tasks | 77 |
| | Scheduled task actions | 79 |
| 18 | Analytics server command-line tools | 83 |
| | Contents | 83 |
| | daemons | 85 |
| | Synopsis | 85 |
| | Options | 85 |
| | runBuilder | 87 |
| | Synopsis | 87 |
| | Description | 87 |
| | Options | 87 |
| | Examples | 88 |
| | runconsole | 89 |
| | Synopsis | 89 |

Part V: Managing Analytics

| | | |
|----|--|----|
| 19 | Secure your data and working environment | 93 |
| | Add user classes and users to Cognos (Windows) | 93 |
| | Add user classes to Cognos (Solaris or Linux) | 95 |
| | Add a new user directly from Upfront | 96 |
| | Open ports to communicate across firewalls | 96 |
| | Modify the database password | 97 |

| | | |
|----|--|-----|
| | Modify the Cognos administrator password | 97 |
| 20 | Manage cubes | 99 |
| | Build a cube manually | 99 |
| | Save a Cognos cube view | 100 |
| | Export a Cognos cube view to a PDF file | 100 |
| | Create a bookmark to a Cognos cube view | 100 |
| | Modify cube properties | 101 |
| 21 | Troubleshooting the server | 103 |
| | Contents | 103 |
| | Check the health of the Analytics server | 103 |
| | Check the Analytics server log files | 104 |
| | Check the status of the Analytics server database connection | 105 |



About Analytics

Introduction to Analytics

Platform Analytics is a workload intelligence solution for LSF cluster, FLEXnet license, and FLEXnet Manager license data. Analytics collects LSF and license data, aggregates it, and transforms it into data cubes for your analysis. Data cubes provide different views and correlations of your data. Analytics provides all the tools you need to collect the data, load it into a database, then transform it for your analysis in an OLAP (online analytical processing) tool.

Architecture overview

The Analytics architecture is based on the Platform Enterprise Reporting Framework (PERF) architecture. Analytics adopts and extends the PERF technology to cover all data collection requirements and to improve data collection reliability.

PERF directories in the Analytics node

PERF components reside in various perf subdirectories within the LSF directory structure. This document uses `LSF_TOP` to refer to the top-level LSF installation directory, and `ANALYTICS_TOP` to refer to the top-level Analytics installation directory. In UNIX, you need to source the PERF environment to use these environment variables.

Table 1: PERF directory environment variables in UNIX

| Directory name | Directory description | Default file path |
|-----------------------------|-----------------------|---------------------------------|
| <code>\$PERF_TOP</code> | PERF directory | <code>ANALYTICS_TOP</code> |
| <code>\$PERF_CONFDIR</code> | Configuration files | <code>ANALYTICS_TOP/conf</code> |
| <code>\$PERF_LOGDIR</code> | Log files | <code>ANALYTICS_TOP/log</code> |
| <code>\$PERF_WORKDIR</code> | Working directory | <code>ANALYTICS_TOP/work</code> |

Table 2: PERF directory environment variables in Windows

| Directory name | Directory description | Default file path |
|-----------------------------|-----------------------|---------------------------------|
| <code>%PERF_TOP%</code> | PERF directory | <code>ANALYTICS_TOP</code> |
| <code>%PERF_CONFDIR%</code> | Configuration files | <code>ANALYTICS_TOP\conf</code> |
| <code>%PERF_LOGDIR%</code> | Log files | <code>ANALYTICS_TOP\log</code> |
| <code>%PERF_WORKDIR%</code> | Working directory | <code>ANALYTICS_TOP\work</code> |

System ports

Analytics hosts use the following TCP ports:

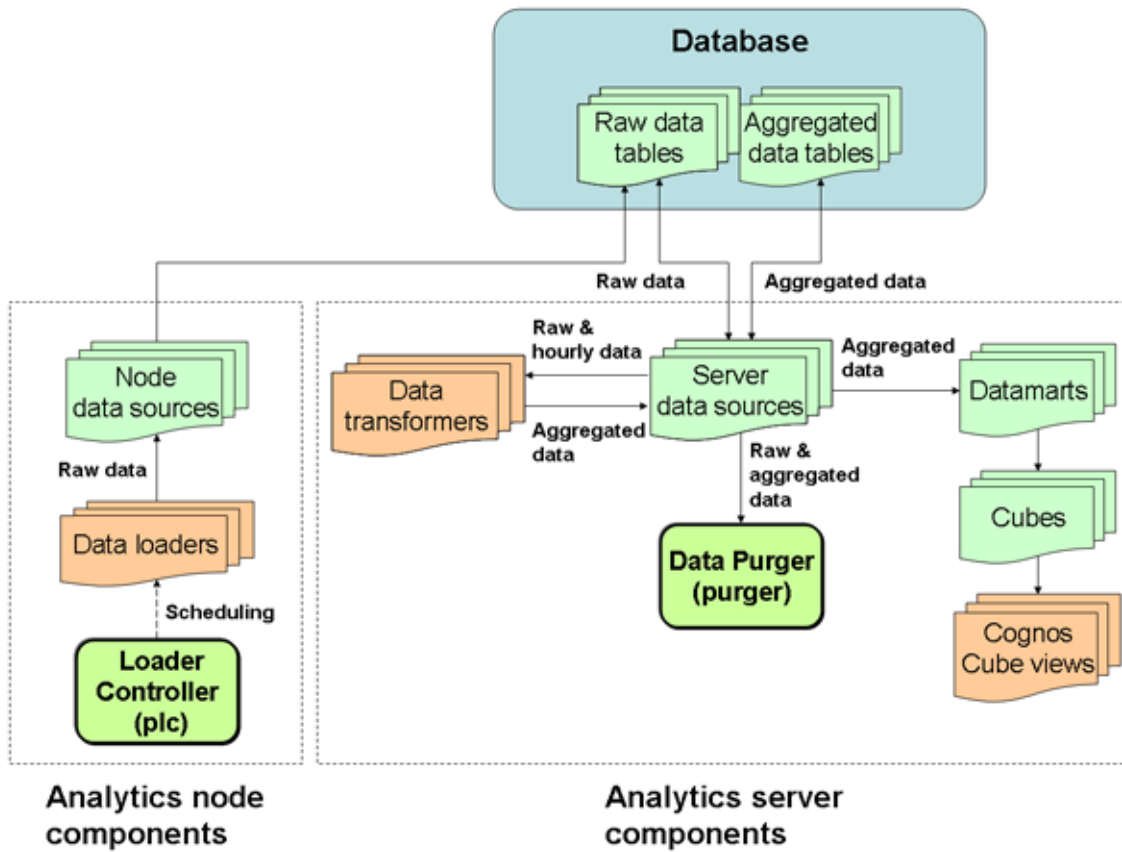
- ◆ The Analytics server requires the use of three consecutive ports, starting with the base port, and are named `PIAD_PORT`, `PIEM_PORT`, and `PIAM_PORT`, respectively. The default base port is 9090 (therefore, the Analytics server uses ports 9090, 9091, and 9092 by default).

PERF directories in the Analytics node

- ❖ The task scheduler uses `PIAD_PORT` (9090 by default) to communicate with the automation manager.
- ❖ The event manager uses `PIEM_PORT` (9091 by default) to receive events from Analytics components.
- ❖ The automation manager uses `PIAM_PORT` (9092 by default) to receive events from Analytics components.
- ◆ The Analytics node requires the use of a port for the loader controller (`plc`) service. The default loader controller port is 4046.
- ◆ The Oracle database host uses 1521 by default.

System architecture

The following diagram illustrates an overview of Analytics architecture.



Licensing

The Analytics license file includes licenses for data collection and cube building. Contact Platform Computing to obtain an Analytics license. You may purchase and enable the following components for your Analytics installation to be included in the Analytics license file:

LSF advanced data collection: The LSF advanced data collection license allows you to collect LSF advanced data from LSF clusters. LSF advanced data is cluster performance and operation data that is not gathered in the base PERF package included with LSF.

License data collection: The license data collection license allows you to collect license usage and event data from your FLEXnet servers.

Cube and datamart building: The cube and datamart building license allows you to build datamarts and cubes.

If you have a demo license and obtained a production license, you need to replace the old demo license file in the *PERF_CONF* directory (*\$PERF_CONF* in UNIX and *%PERF_CONF%* in Windows) with the new demo license file. Make sure that both licenses have the same file name (usually *license.dat*).

II

Database host

Database

The relational database contains the cluster data, organised into tables, for reporting and analysis.

About the database

The relational database contains the cluster operations data for reporting and analysis. Analytics supports Oracle 9i and Oracle 10g databases. Analytics components input and output data from the tables within the database.

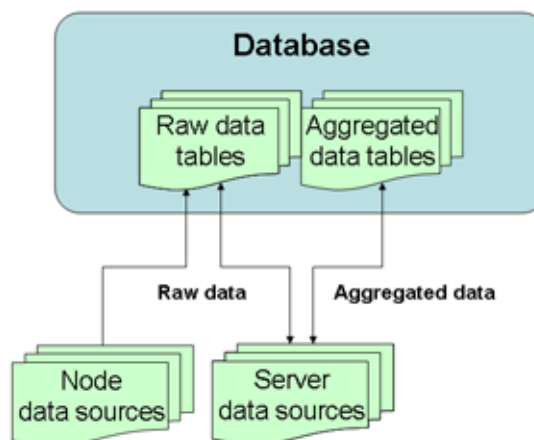
Default behavior

Data is stored and organized in tables within the database. The organization of this data is defined in the data schema of the tables.

Database interactions

All interactions between Analytics and the database are through the JDBC connection as defined by the data sources. The Analytics node interacts with the raw data tables in the database, while the Analytics server interacts with both the raw and aggregated data tables.

The following diagram illustrates the interaction between the database and other components.



About the database

Data sources

Data sources define the JDBC connections between the hosts and the database.

Contents

- ◆ [About data sources](#) on page 21
- ◆ [Data source actions](#) on page 22

About data sources

Data sources define all JDBC connections to the raw and aggregated data tables in the relational database. The data tables contain processed cluster data that can be extracted and used to build datamarts.

You define the JDBC connection to the database when you install Analytics. The information about the JDBC driver together with the user and password information is called the data source. If you change your database or modify your connection, you need to update the data source properties in Analytics accordingly. The default Analytics data source for the server and the node is `ReportDB`.

Analytics uses one or more data sources. Multiple datamarts can use a single data source. You must install JDBC drivers for your database type on the Analytics server host before defining the corresponding data source.

Data source interactions

The data source is the JDBC connection between the data tables in the relational database and all Analytics components. Any interaction with the data tables in the database goes through the JDBC connection as defined in the data source.

Server data source interactions

Data transformers obtain raw data and aggregated hourly data from the data tables through the server data sources, and stores aggregated data (including hourly and daily data) into the data tables through the server data sources.

The data purger purges old records from the raw and aggregated data tables through the server data sources.

Datamarts obtain data from the aggregated data tables through the server data sources.

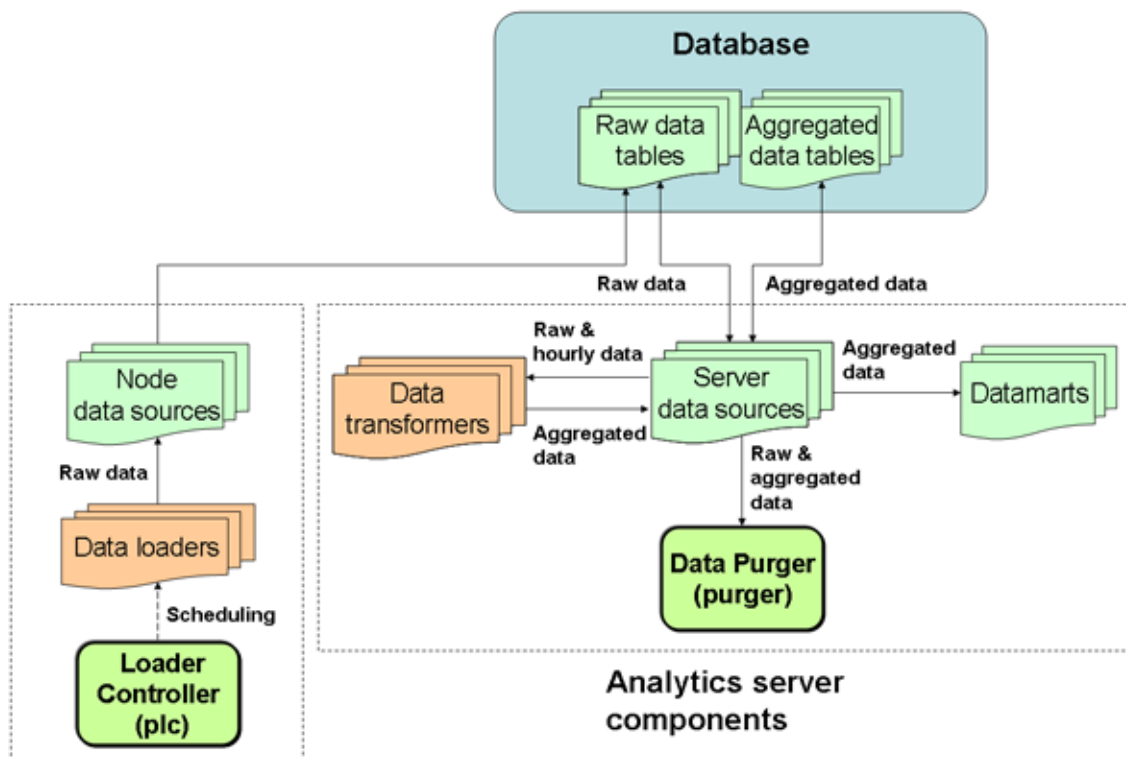
Data source actions

Node data source interactions

The data sources for the Analytics node interact with the raw data tables in the database, while the data sources for the Analytics server interact with both the raw and the aggregated data tables in the database. If your cluster has multiple FLEXnet Manager servers, each FLEXnet Manager server has its own data source.

Data loaders either request cluster operation data, or obtain it directly from the data tables through the node data sources. The data loaders store this data into raw data tables through the node data sources.

The following diagram illustrates the interaction between data sources and other components.



Data source actions

Actions on the Analytics server data sources

If the Analytics server is running on a UNIX (Solaris or Linux) host, you need to restart the Analytics server daemons (by running `daemons.sh stop` and `daemons.sh start`) after changing the server data source.

| Action | Analytics Console |
|--|---|
| View the list of server data sources. | In the navigation tree, click Data Sources . |
| Add a server data source. | When viewing the list of data sources, select Action > Add Data Source . |
| Edit the settings of a server data source. | When viewing the list of data sources, click the data source and select Action > Edit Data Source . |
| Delete a server data source. | When viewing the list of data sources, click the data source and select Action > Remove Data Source . |

Actions on the Analytics node data sources

| Action | Command line |
|--|--|
| Add a node data source. | UNIX (Solaris or Linux): <code>dbconfig.sh add <i>data_source_name</i></code> Windows: <code>dbconfig add <i>data_source_name</i></code> where ◆ <i>data_source_name</i> is the name the data source that you want to add. |
| Edit the settings of the Analytics node data source (ReportDB). | UNIX (Solaris or Linux): <code>dbconfig.sh</code> Windows: <code>dbconfig</code> |
| Edit the settings of any node data source, including FLEXnet Manager data sources. | UNIX (Solaris or Linux): <code>dbconfig.sh edit <i>data_source_name</i></code> Windows: <code>dbconfig edit <i>data_source_name</i></code> where ◆ <i>data_source_name</i> is the name the data source that you want to edit. |

Data source actions

III

Analytics node

Analytics nodes are hosts that collect data from clusters or license servers. Each node either belongs to a cluster from which Analytics collects data, or is a standalone host that collects license data.

Loader controller

The Platform loader controller (plc) controls the data loaders that gather data from the system and writes the data into the relational database containing raw data.

Contents

- ◆ [About the loader controller](#) on page 27
- ◆ [Configuration to modify loader controller behavior](#) on page 28
- ◆ [Loader controller actions](#) on page 29

About the loader controller

The loader controller manages the data loaders by controlling the schedule in which each data loader gathers data.

Logging levels

There are logging levels that determine the detail of messages that the PERF services record in the log files. In decreasing level of detail, these levels are ALL (all messages), TRACE, DEBUG, INFO, WARN, ERROR, FATAL, and OFF (no messages).

By default, the PERF services log messages of INFO level or higher (that is, all INFO, WARN, ERROR, and FATAL messages).

The loader controller log file is located in the log directory:

- ◆ UNIX (Solaris or Linux): `$PERF_LOGDIR`
- ◆ Windows: `%PERF_LOGDIR%`

Default behavior

The loader controller service starts automatically when the master host starts up if you have the loader controller registered as an RC.

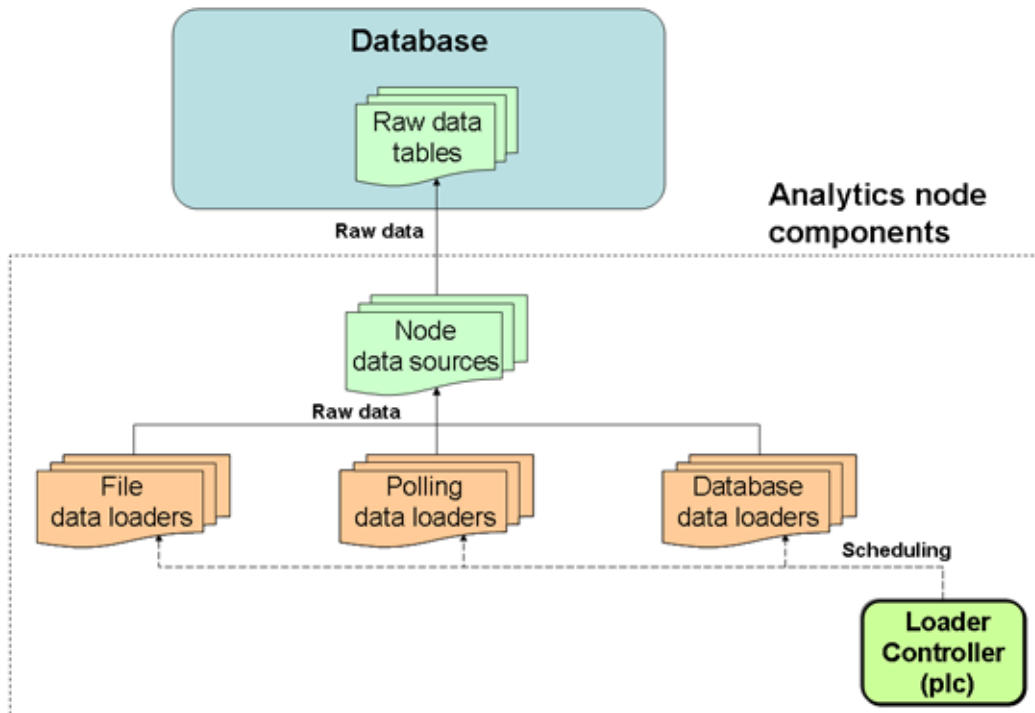
Loader controller interactions

The loader controller service controls the scheduling of the data loaders. Sampling and retrieving data loaders request cluster operation data from the raw data tables through the node data sources while other data loaders obtain it directly from the

Configuration to modify loader controller behavior

raw data tables through the node data sources. The data loaders store this data into raw data tables through the node data sources. Each data loader contains data that is stored in specific raw data tables in the database.

The following diagram illustrates the interaction between the loader controller and other components.



Configuration to modify loader controller behavior

| Action | Configuration files | Parameter and syntax |
|---|---|---|
| Specify the default log level of your plc log file. | log4j.properties File location: UNIX: \$PERF_CONFDIR Windows: %PERF_CONFDIR% | log4j.logger.com.platform.perf.dataloader= <i>log_level</i> , com.platform.perf.dataloader where <ul style="list-style-type: none"> <i>log_level</i> is the default log level of your loader controller log files. The loader controller only logs messages of the same or lower level of detail as <i>log_level</i> . Therefore, if you change the log level to ERROR, the loader controller will only log ERROR and FATAL messages. |

Configuration to modify loader controller behavior for EGO-enabled clusters only

After editing the loader controller service configuration file (`plc_service.xml`), restart the loader controller and LSF on the master host for your changes to take effect.

| Action | Configuration files | Parameter and syntax |
|---|--|---|
| Enable automatic startup of the <code>plc</code> service. This is the default behavior. | <code>plc_service.xml</code> File location: | <code><sc:StartType>AUTOMATIC</sc:StartType></code> |
| Disable automatic startup of the <code>plc</code> service. | UNIX: <code>\$LSF_ENVDIR/perf/cluster_name/conf</code> Windows: <code>%LSF_ENVDIR%\perf\cluster_name\conf</code> | <code><sc:StartType>MANUAL</sc:StartType></code> |

Loader controller actions

Actions on the loader controller service

NOTE: To stop or start the `plc` service, you must run the commands on the local host running the `plc` service.

| Action | Command line |
|--|----------------------------------|
| View the status of the <code>plc</code> and other PERF services. | <code>perfadmin list</code> |
| Stop the <code>plc</code> service. | <code>perfadmin stop plc</code> |
| Start the <code>plc</code> service. | <code>perfadmin start plc</code> |

Actions to change the loader controller settings

| Action | Command line |
|--|--|
| Dynamically change the log level of your loader controller log file (temporarily). | UNIX (Solaris or Linux): <code>plcclient.sh -l <i>log_level</i></code> Windows: <code>plcclient -l <i>log_level</i></code> where <ul style="list-style-type: none"> ◆ <i>log_level</i> is the log level of your loader controller log file. If you restart the loader controller, these settings will revert back to the default level. <p>NOTE: You must run this command on the local host running the <code>plc</code> service.</p> |

Loader controller actions

Data loaders

Data loaders gather cluster operation data and load it into tables in a relational database containing raw data. Data loaders are controlled by the Platform loader controller (plc) service.

Contents

- ◆ [About data loaders](#) on page 31
- ◆ [Configuration to modify data loader behavior](#) on page 33
- ◆ [Data loader actions](#) on page 34

About data loaders

Data loaders are polling loaders or history data loaders. The data loaders gather data and load this data into specific tables in the relational database containing raw data. Data loaders handle daylight savings automatically by using GMT time when gathering data.

Logging levels

There are logging levels that determine the detail of messages that the data loaders record in the log files. In decreasing level of detail, these levels are ALL (all messages), TRACE, DEBUG, INFO, WARN, ERROR, FATAL, and OFF (no messages).

By default, data loaders log messages of INFO level or higher (that is, all INFO, WARN, ERROR, and FATAL messages).

The data loader log files are located in the `data_loader` subdirectory of the log directory:

- ◆ UNIX (Solaris or Linux): `$PERF_LOGDIR/data_loader`
- ◆ Windows: `%PERF_LOGDIR%\data_loader`

Default behavior

Data loaders gather data from data sources at regular intervals. The following are lists of the data loaders and default behavior:

About data loaders

Table 3: FLEXnet data loaders

| Data loader name | Data type | Data gathering interval | Data loads to | Loader type |
|--|------------------------|-------------------------|------------------------|-------------|
| FLEXnet usage (flexlicusageloader) | license usage | 5 minutes | FLEXLM_LICENSE_USAGE | polling |
| FLEXnet events (flexliceventsloader) | license log file event | 5 minutes | FLEXLM_LICENSE_EVENTS | file |
| FLEXnet Manager (fnmloader) - only supports FLEXnet Manager 11 or later. | license event | 30 minutes | FLEXNET_LICENSE_EVENTS | database |

Table 4: LSF data loaders

| Data loader name | Data type | Data gathering interval | Data loads to | Loader type |
|---|---|-------------------------|---|-------------|
| Host metrics (hostmetricsloader) | host-related metrics | 10 minutes | RESOURCE_METRICS RESOURCES_RESOURCE_METRICS | polling |
| Host properties (hostpropertiesloader) | resource properties | 1 hour | LSF_RESOURCE_PROPERTIES | polling |
| Bhosts (lsfbhostsloader) | host utilization and state-related | 5 minutes | LSF_BHOSTS | polling |
| LSF events (lsfeventsloader) | events with a job ID, performance events, resource events | 5 minutes | LSB_EVENTS LSB_EVENTS_EXECHOSTLIST LSF_PERFORMANCE_METRIC | file |
| Resource properties (lsfresproploader) | shared resource properties | 1 hour | LSF_RESOURCE_PROPERTIES | polling |
| SLA (lsfslaloader) | SLA performance | 5 minutes | LSF_SLA | polling |
| Shared resource usage (sharedresusageloader) | shared resource usage | 5 minutes | SHARED_RESOURCE_USAGE SHARED_RESOURCE_USAGE_HOST LIST | polling |

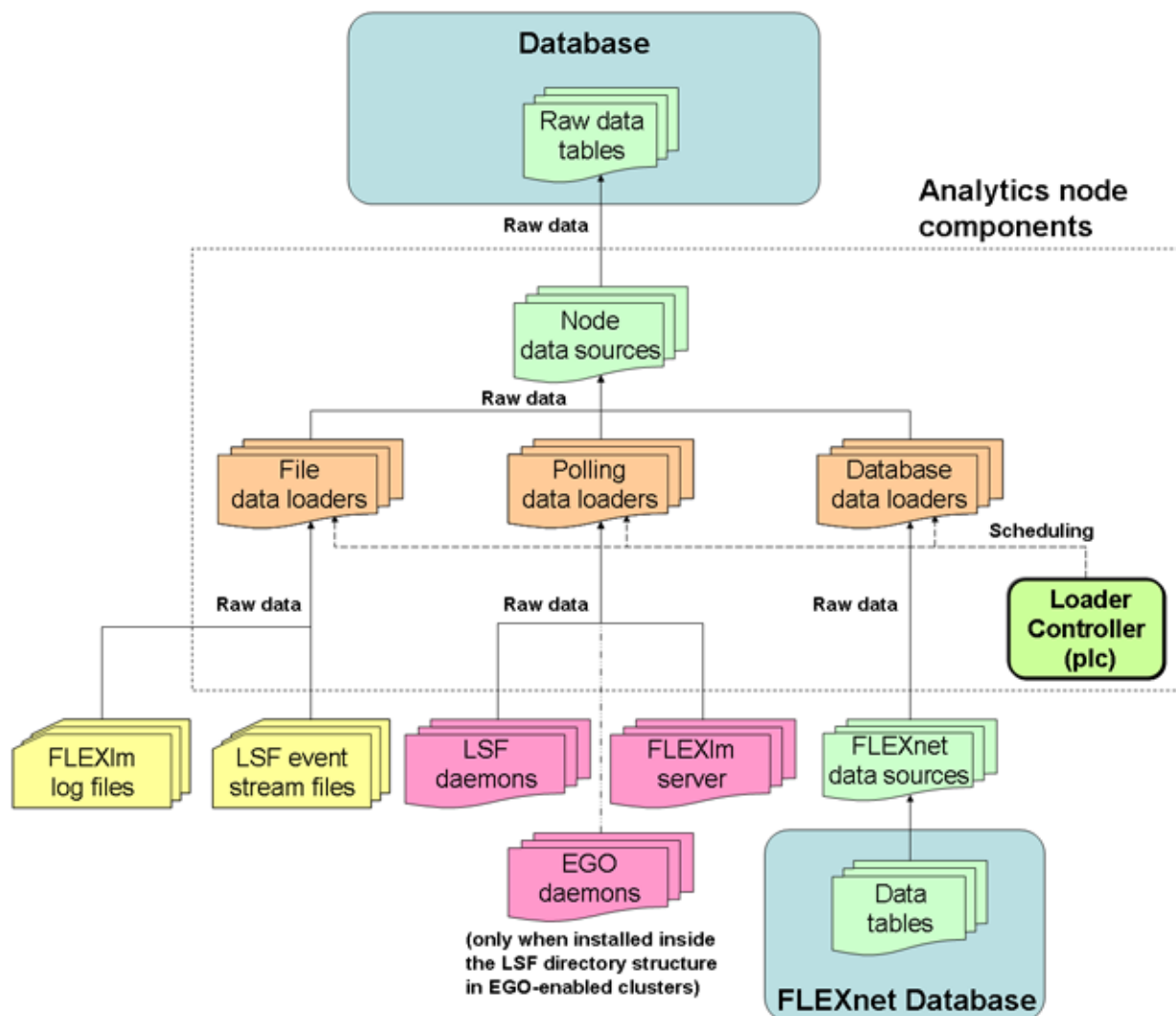
Table 5: LSF advanced data loaders

| Data loader name | Data type | Data gathering interval | Data loads to | Loader type |
|--|---------------------|-------------------------|---------------------------------------|-------------|
| Host group (hostgrouploader) | host group | 1 hour | HOST_GROUP | polling |
| Bqueues (lsfbqueueloader) | queue properties | 5 minutes | LSF_BQUEUES | polling |
| Pending reason (lsfpendingreasonloader) | job pending reasons | 15 minutes | JOBS_PENDING_REASON DPR_BYINTERVAL | polling |
| User group (usergrouploader) | user group | 1 hour | USER_GROUP | polling |

Data loader interactions

The loader controller service controls the scheduling of the data loaders. The data loaders store LSF and license data into raw data tables through the node data sources. Each data loader contains data that is stored in specific raw data tables in the database.

The following diagram illustrates the interaction between data loaders and other components.



Configuration to modify data loader behavior

After editing the loader controller configuration files, restart the loader controller for your changes to take effect. The specific loader controller configuration file depends on the type of data loader:

plc_license.xml: FLEXnet data loaders

plc_lsfx.xml: LSF data loaders

plc_lsfx_advanced.xml: LSF advanced data loaders

These files are located in the loader controller configuration directory:

- ❖ UNIX (Solaris or Linux): \$PERF_CONFDIR/plc
- ❖ Windows: %PERF_CONFDIR%\plc

Data loader actions

| Action | Configuration files | Parameter and syntax |
|--|---|---|
| Specify the frequency of data gathering for the specified data loader. | Loader controller configuration files for your data loaders (plc_*.xml). | <pre><DataLoader Name="loader_name" Interval="gather_interval" ... /></pre> <p>where</p> <ul style="list-style-type: none"> ◆ <i>loader_name</i> is the name of your data loader ◆ <i>gather_interval</i> is the time interval between data gathering, in seconds |
| Enable data gathering for the specified data loader. This is the default behavior. | | <pre><DataLoader Name="loader_name" ... Enable="true" ... /></pre> <p>where</p> <ul style="list-style-type: none"> ◆ <i>loader_name</i> is the name of your data loader |
| Disable data gathering for the specified data loader. | | <pre><DataLoader Name="loader_name" ... Enable="false" ... /></pre> <p>where</p> <ul style="list-style-type: none"> ◆ <i>loader_name</i> is the name of your data loader |
| Enable data loss protection for the specified data loader. This is the default behavior. | Specific data loader configuration file: <i>dataloader_name.xml</i> | <pre><Writer ... EnableRecover="Y"></pre> |
| Disable data loss protection for the specified data loader. | File location: UNIX: \$PERF_CONFDIR/dataloader Windows: %PERF_CONFDIR%\dataloader | <pre><Writer ... EnableRecover="N"></pre> |
| Specify the default log level of your data loader log files. | log4j.properties File location: UNIX: \$PERF_CONFDIR Windows: %PERF_CONFDIR% | <pre>log4j.logger.\${dataloader}=log_level, \${dataloader}</pre> <p>where</p> <ul style="list-style-type: none"> ◆ <i>log_level</i> is the default log level of your data loader log files. <p>The data loaders only log messages of the same or lower level of detail as <i>log_level</i>. Therefore, if you change the log level to <code>ERROR</code>, the data loaders will only log <code>ERROR</code> and <code>FATAL</code> messages.</p> |

Data loader actions

Actions to view the data loader status

| Action | Command line |
|--------------------------------------|---|
| View the status of the data loaders. | UNIX (Solaris or Linux): <code>plcclient.sh -s</code> Windows: <code>plcclient -s</code> |

Actions to change the data loader settings

| Action | Command line |
|---|---|
| Dynamically change the log level of your data loader log files (temporarily). | UNIX (Solaris or Linux): <code>plcclient.sh -n loader_name -l log_level</code> Windows: <code>plcclient -n loader_name -l log_level</code> where <ul style="list-style-type: none"> ◆ <i>loader_name</i> is the name of your data loader ◆ <i>log_level</i> is the log level of your data loader log files. If you restart the loader controller, these settings will revert back to the default level. |

View or dynamically edit the data loader settings

Use the Analytics Console to view or edit the data loader settings. Any changes you make to the settings are permanent (that is, even after restarting the loader controller).

- 1 In the navigation tree of the Analytics Console, select **Data Collection Nodes**.
- 2 Right-click the loader controller for your cluster and select **Loader properties**.

NOTE: You can only view the data loader properties when the corresponding loader controller is running.

- 3 Right-click the data loader you want to view or edit and select **Properties**.
- 4 Edit the data loader parameters, if needed.
You can edit the following data loader parameters:
 - ◆ **Parameters:** The specific parameters for the data loader. You can only edit the parameters of FLEXnet data loaders (`flexlicusageloader` and `flexliceventsloader`).
 - ◆ **Interval (seconds):** The data gathering interval of the data loader, in seconds.
 - ◆ **Log level:** The data loader logs messages of a level specified here and higher.
 - ◆ **Description:** A description of the data loader.
- 5 To save any changes and close the window, click **OK**.

Data loader actions

Analytics node command-line tools

Contents

- ◆ [dbconfig](#) on page 39
- ◆ [perfadmin](#) on page 41
- ◆ [plclient](#) on page 43

Contents

dbconfig

Configures the node data source.

Synopsis

```
dbconfig.sh [add data_source_name | edit data_source_name]
```

```
dbconfig [add data_source_name | edit data_source_name]
```

```
dbconfig.sh -h
```

```
dbconfig -h
```

`dbconfig.sh` is the command for UNIX (Solaris or Linux) and `dbconfig` is the command for Windows.

Description

Run the command to configure the Analytics node data source (ReportDB).

If you are running this command locally from an Analytics node running UNIX (Solaris or Linux), you need to be running X-Windows. If you are running this command remotely, you need to set your display environment.

Options

add *data_source_name*

Adds the specified data source to the Analytics node.

edit *data_source_name*

Edits the specified data source on the Analytics node.

-h

Prints the command usage and exits.

Options

perfadmin

Administer the PERF services.

Synopsis

```
perfadmin start service_name | all
perfadmin stop service_name | all
perfadmin [list | -h]
```

Description

Starts or stops the PERF services, or shows status.

Run the command on the Analytics node to control the loader controller service (plc).

Options

start *service_name* | all

Starts the PERF services on the local host. You must specify the service name or the keyword `all`. Do not run this command on a host that is not the Analytics node, you should only run one set of services per cluster.

stop *service_name* | all

Stops the PERF services on the local host. You must specify the service name or the keyword `all`.

list Lists status of PERF services. Run this command on the PERF host.

-h Outputs command usage and exits.

Output

Status information and prompts are displayed in your command console.

SERVICE The name of the PERF service.

STATUS STARTED

Service is running.

STOPPED

Service is not running.

UNKNOWN

Service status is unknown. The local host may not be the PERF host.

WSM_PID Process ID of the running service.

HOST_NAME Name of the host.

Output

plcclient

Administer the loader controller or data loaders.

Synopsis

```
plcclient.sh [-s]
```

```
plcclient [-s]
```

```
plcclient.sh [-l log_level]
```

```
plcclient [-l log_level]
```

```
plcclient.sh [-n loader_name -l log_level]
```

```
plcclient [-n loader_name -l log_level]
```

`plcclient.sh` is the command for UNIX (Solaris or Linux) and `plcclient` is the command for Windows.

Description

Run the command to administer the loader controller or the data loaders.

Options

-s

View the status of the data loaders.

-l *log_level*

Dynamically change the log level of the loader controller to the specified log level. If you restart the loader controller (`plc`) service, this setting will revert back to the default level.

-n *loader_name* -l *log_level*

Dynamically change the log level of the specified data loader to the specified log level. If you restart the loader controller (`plc`) service, this setting will revert back to the default level.

Options

Analytics node configuration files

Contents

- ◆ [perf.conf](#) on page 47

Contents

perf.conf

The `perf.conf` file controls the operation of PERF.

About perf.conf

`perf.conf` specifies the version and configuration of various PERF components and features. The `perf.conf` file also specifies the file path to PERF directories and the PERF license file.

The `perf.conf` file is used by Analytics and applications built on top of it. For example, information in `perf.conf` is used by Analytics daemons and commands to locate other configuration files, executables, and services. `perf.conf` is updated, if necessary, when you upgrade to a new version of Analytics.

Changing perf.conf configuration

After making any changes to `perf.conf`, run the following commands to restart the PERF services and apply your changes:

```
perfadmin stop all
perfadmin start all
```

Location

The default location of `perf.conf` is in `/conf`. This default location can be overridden if necessary by the environment variable `PERF_CONFDIR`.

Format

Each entry in `perf.conf` has the following form:

```
NAME=VALUE
```

The equal sign `=` must follow each `NAME` and there should be no space beside the equal sign. Text starting with a pound sign (`#`) are comments and are ignored. Do not use `#i f` as this is reserved syntax for time-based configuration.

DLP_ENABLED

Syntax

```
DLP_ENABLED=Y | N
```

Description

Enables data loss protection (DLP) for data loaders. If enabled, you can enable or disable data loss protection for specific data loaders in the Analytics node by editing the specific data loader configuration file. If disabled, data loss protection is disabled in all data loaders in the Analytics node and cannot be enabled in the specific data loader configuration file.

Default

Y (Enabled). In addition, all data loaders have data loss protection enabled by default.

EGO_VERSION

EGO_VERSION

Syntax

```
EGO_VERSION=version_number
```

Description

Specifies the version of EGO in the LSF cluster to which the Analytics node belongs.

Example

```
EGO_VERSION=1.2
```

Default

By default, EGO_VERSION is set to the version of EGO in the LSF cluster to which the Analytics node belongs.

LICENSE_FILE

Syntax

```
LICENSE_FILE="file_name ... |  
port_number@host_name[:port_number@host_name ...]"
```

Description

Specifies one or more demo or permanent license files used by Analytics.

The value for LICENSE_FILE can be either of the following:

- ◆ The full path name to the license file.
 - ❖ UNIX example:

```
LICENSE_FILE=/usr/share/lsf/cluster1/conf/license.dat
```
 - ❖ Windows examples:

```
LICENSE_FILE= C:\\licenses\\license.dat  
LICENSE_FILE=\\HostA\\licenses\\license.dat
```
 - ◆ For a permanent license, the name of the license server host and TCP port number used by the `lmgrd` daemon, in the format `port@host_name`. For example:

```
LICENSE_FILE="1700@hostD"
```
 - ◆ For a license with redundant servers, use a comma to separate the `port@host_names`. The port number must be the same as that specified in the `SERVER` line of the license file. For example:
UNIX:

```
LICENSE_FILE="port@hostA:port@hostB:port@hostC"
```


Windows:

```
LICENSE_FILE="port@hostA;port@hostB;port@hostC"
```
- Multiple license files should be quoted and must be separated by a pipe character (`|`).

Windows example:

```
LICENSE_FILE="C:\licenses\license1|C:\licenses\license2|D:\mydir\license3"
```

Multiple files may be kept in the same directory, but each one must reference a different license server. When checking out a license, Analytics searches the servers in the order in which they are listed, so it checks the second server when there are no more licenses available from the first server.

If this parameter is not defined, Analytics assumes the default location.

Default

By default, LICENSE_FILE is set as the file path to the license file that you specified during the initial Analytics installation.

If you installed FLEXlm separately from Analytics to manage other software licenses, the default FLEXlm installation puts the license file in the following location:

- ◆ UNIX: /usr/share/flexlm/licenses/license.dat
- ◆ Windows: C:\flexlm\license.dat

LICENSE_VERSION

Syntax

```
LICENSE_VERSION=version_number
```

Description

Specifies the version of the license module installed with Analytics.

Example

```
LICENSE_VERSION=7.0
```

Default

Not defined.

LOADER_BATCH_SIZE

Syntax

```
LOADER_BATCH_SIZE=integer
```

Description

Specifies the number of SQL statements that can be submitted to Oracle at the same time.

Valid values

Any positive, non-zero integer.

Default

5000

LSF_ENVDIR

LSF_ENVDIR

Syntax

`LSF_ENVDIR=directory`

Description

Specifies the LSF configuration directory, which is the directory containing the `lsf.conf` file.

Default

`/etc`

LSF_VERSION

Syntax

`LSF_VERSION=version_number`

Description

Specifies the version of LSF in the cluster to which the Analytics node belongs.

Example

`LSF_VERSION=7.0`

Default

By default, `LSF_VERSION` is set to the version of LSF in the cluster to which the Analytics node belongs.

PERF_CONFDIR

Syntax

`PERF_CONFDIR=directory`

Description

Specifies the configuration directory, which contains the configuration files for Analytics node components.

Default

- ◆ UNIX: `ANALYTICS_TOP/conf`
- ◆ Windows: `ANALYTICS_TOP\conf`

where `ANALYTICS_TOP` is the top-level Analytics node installation directory.

PERF_LOGDIR

Syntax

`PERF_LOGDIR=directory`

Description

Specifies the logging directory, which contains the log files for Analytics node components.

Default

- ◆ UNIX: *ANALYTICS_TOP*/log
- ◆ Windows: *ANALYTICS_TOP*\log

where *ANALYTICS_TOP* is the top-level Analytics node installation directory.

PERF_TOP

Syntax

PERF_TOP=*directory*

Description

Specifies the top-level PERF directory.

Default

- ◆ UNIX: *ANALYTICS_TOP*
- ◆ Windows: *ANALYTICS_TOP*

where *ANALYTICS_TOP* is the top-level Analytics node installation directory.

PERF_VERSION

Syntax

PERF_VERSION=*version_number*

Description

Specifies the version of PERF installed with the Analytics node.

Example

PERF_VERSION=1.2.3

Default

Not defined.

PERF_WORKDIR

Syntax

PERF_WORKDIR=*directory*

Description

Specifies the working directory.

Default

- ◆ UNIX: *ANALYTICS_TOP*/work

PERF_WORKDIR

◆ **Windows:** *ANALYTICS_TOP*\work

where *ANALYTICS_TOP* is the top-level Analytics node installation directory

Troubleshooting the node

Contents

- ◆ [Change the default log level of your log files](#) on page 53
- ◆ [Disable data collection for individual data loaders](#) on page 54
- ◆ [Check the status of the loader controller](#) on page 55
- ◆ [Check the status of the data loaders](#) on page 55

Change the default log level of your log files

Change the default log level of your log files if they do not cover enough detail, or cover too much, to suit your needs.

- 1 If you are logged into a UNIX (Solaris or Linux) host, source the LSF environment.
 - ◆ For `csh` or `tcsh`: `source LSF_TOP/conf/cshrc.lsf`
 - ◆ For `sh`, `ksh`, or `bash`: `. LSF_TOP/conf/profile.lsf`
- 2 If you are logged into a UNIX (Solaris or Linux) host, source the PERF environment.
 - ◆ For `csh` or `tcsh`: `source $PERF_TOP/conf/cshrc.perf`
 - ◆ For `sh`, `ksh`, or `bash`: `. $PERF_TOP/conf/profile.perf`
- 3 Edit the `log4j.properties` file.

This file is located in the PERF configuration directory:

 - ◆ UNIX (Solaris or Linux): `$PERF_CONFDIR`
 - ◆ Windows: `%PERF_CONFDIR%`
- 4 Navigate to the section representing the service you want to change, or to the default loader configuration if you want to change the log level of the data loaders, and look for the `*.logger.*` variable.

For example, to change the log level of the loader controller log files, navigate to the following section, which is set to the default `INFO` level:

```
# Loader controller ("plc") configuration
```

Disable data collection for individual data loaders

```
log4j.logger.com.platform.perf.dataloader=INFO
com.platform.perf.dataloader
```

- 5 Change the `*.logger.*` variable to the new logging level.

In decreasing level of detail, the valid values are ALL (for all messages), DEBUG, INFO, WARN, ERROR, FATAL, and OFF (for no messages). The services or data loaders only log messages of the same or lower level of detail as specified by the `*.logger.*` variable. Therefore, if you change the log level to ERROR, the service or data loaders will only log ERROR and FATAL messages.

For example, to change the loader controller log files to the ERROR log level:

```
# Loader controller ("plc") configuration
log4j.logger.com.platform.perf.dataloader=ERROR
com.platform.perf.dataloader
```

- 6 Restart the service that you changed (or the loader controller if you changed the data loader log level).

Disable data collection for individual data loaders

To reduce unwanted data from being logged in the database, disable data collection for individual data loaders.

- 1 If you are logged into a UNIX (Solaris or Linux) host, source the LSF environment.
 - ❖ For `csh` or `tcsh`: `source LSF_TOP/conf/cshrc.lsf`
 - ❖ For `sh`, `ksh`, or `bash`: `. LSF_TOP/conf/profile.lsf`
- 2 If you are logged into a UNIX (Solaris or Linux) host, source the PERF environment.
 - ❖ For `csh` or `tcsh`: `source $PERF_TOP/conf/cshrc.perf`
 - ❖ For `sh`, `ksh`, or `bash`: `. $PERF_TOP/conf/profile.perf`
- 3 Edit the `plc` configuration files for your data loaders.
 - ❖ For LSF data loaders, edit `plc_lsf.xml`.
 - ❖ For FLEXnet data loaders, edit `plc_license.xml`.
 - ❖ For LSF advanced data loaders, edit `plc_lsf_advanced_data.xml`.

These files are located in the LSF environment directory:

 - ◆ Unix (Solaris or Linux): `$LSF_ENVDIR`
 - ◆ Windows: `%LSF_ENVDIR%`
- 4 Navigate to the specific `<DataLoader>` tag with the `Name` attribute matching the data loader that you want to disable.

For example:

```
<DataLoader Name="hostgrouploader" ... Enable="true" .../>
```
- 5 Edit the `Enable` attribute to `"false"`.

For example, to disable data collection for this plug-in:

```
<DataLoader Name="hostgrouploader" ... Enable="false" ... />
```

- 6 Restart the `plc` service.
-

Check the status of the loader controller

- 1 If you are logged into a UNIX (Solaris or Linux) host, source the LSF environment.
 - ❖ For `csh` or `tcsh`: `source LSF_TOP/conf/cshrc.lsf`
 - ❖ For `sh`, `ksh`, or `bash`: `. LSF_TOP/conf/profile.lsf`
 - 2 If you are logged into a UNIX (Solaris or Linux) host, source the PERF environment.
 - ❖ For `csh` or `tcsh`: `source $PERF_TOP/conf/cshrc.perf`
 - ❖ For `sh`, `ksh`, or `bash`: `. $PERF_TOP/conf/profile.perf`
 - 3 Navigate to the PERF binary directory.
 - ❖ UNIX (Solaris or Linux): `cd $PERF_TOP/version_number/bin`
 - ❖ Windows: `cd %PERF_TOP%\version_number\bin`
 - 4 View the status of the loader controller (`plc`) and other PERF services.


```
perfadmin list
```
 - 5 Verify that there are no errors in the loader controller log file.

The loader controller log file is located in the log directory:

 - ◆ UNIX (Solaris or Linux): `$PERF_LOGDIR`
 - ◆ Windows: `%PERF_LOGDIR%`
-

Check the status of the data loaders

- 1 If you are logged into a UNIX (Solaris or Linux) host, source the LSF environment.
 - ❖ For `csh` or `tcsh`: `source LSF_TOP/conf/cshrc.lsf`
 - ❖ For `sh`, `ksh`, or `bash`: `. LSF_TOP/conf/profile.lsf`
- 2 If you are logged into a UNIX (Solaris or Linux) host, source the PERF environment.
 - ❖ For `csh` or `tcsh`: `source $PERF_TOP/conf/cshrc.perf`
 - ❖ For `sh`, `ksh`, or `bash`: `. $PERF_TOP/conf/profile.perf`
- 3 Verify that there are no errors in the LSF data loader log files.

The data loader log files (`data_loader_name.log.host_name`) are located in the `dataloader` subdirectory of the log directory:

 - ◆ UNIX (Solaris or Linux): `$PERF_LOGDIR/dataloader`

Check the status of the Analytics node database connection

- ◆ Windows: %PERF_LOGDIR%\dataloader
-

Check the status of the Analytics node database connection

- 1 If you are logged into a UNIX (Solaris or Linux) host, source the LSF environment.
 - ◆ For `csh` or `tcsh`: `source LSF_TOP/conf/cshrc.lsf`
 - ◆ For `sh`, `ksh`, or `bash`: `. LSF_TOP/conf/profile.lsf`
 - 2 If you are logged into a UNIX (Solaris or Linux) host, source the PERF environment.
 - ◆ For `csh` or `tcsh`: `source $PERF_TOP/conf/cshrc.perf`
 - ◆ For `sh`, `ksh`, or `bash`: `. $PERF_TOP/conf/profile.perf`
 - 3 Navigate to the PERF binary directory.
 - ◆ UNIX (Solaris or Linux): `cd $PERF_TOP/version_number/bin`
 - ◆ Windows: `cd %PERF_TOP%\version_number\bin`
 - 4 View the status of the node database connection.
 - ◆ UNIX (Solaris or Linux): `dbconfig.sh`
 - ◆ Windows: `dbconfig`
-

IV

Analytics server

The Analytics server manages the data that the Analytics nodes collect. You can perform all server functions using the Analytics Console in the Analytics server.

The server performs the following functions:

- ◆ Analytics node management
- ◆ Cluster data management
- ◆ Reporting

Analytics Console

The Platform Analytics Console allows you to view cluster data and Analytics configuration.

Contents

- ◆ [About the Analytics Console](#) on page 59
- ◆ [Analytics Console actions](#) on page 60

About the Analytics Console

The Platform Analytics Console displays information on your cluster and Analytics configuration. You can also make some configuration changes to Analytics components. You can view the following data in the Analytics Console:

- Datamart ETL flows** Displays the information and status of each datamart ETL flow in the system.
- Cubes** Displays the information and status of each cube in the system.
- Data mappings** A data mapping file enables Analytics to substitute different names for the data in a table. The mapping substitution occurs when Analytics builds the datamart or transforms data.
- Clusters** Displays information on each cluster that Analytics monitors.
- Data collection nodes** This includes all Analytics nodes in the system.
- Data sources** This includes the data sources that are running on the Analytics server and nodes.
- Scheduled tasks** This includes the status and schedule of all scheduled tasks that the Analytics server controls, including the datamart building, data purger, cube builder, and data transformers.
- Events** Displays each event logged in Analytics. You can filter the display of these events to find specific events.
- Data purger** Displays information about each data purger task.

Analytics Console actions

| Action | Command line |
|------------------------------|---|
| Start the Analytics Console. | <p>UNIX (Solaris or Linux): <code>ANALYTICS_TOP/bin/runconsole.sh</code></p> <p>Windows: Start > Programs > Platform Analytics <i>version_number</i> Server > Platform Analytics <i>version_number</i> Console</p> <p>If you are running this command locally from the Analytics server running UNIX (Solaris or Linux), you need to be running X-Windows. If you are running this command remotely, you need to set your display environment.</p> <hr/> <p>IMPORTANT: The Analytics server must have access to the Analytics data source (ReportDB). If the Analytics server cannot connect to the data source, the data source configuration tool displays and the Analytics Console will not start up until you can connect to the data source.</p> <hr/> |

Data transformers

Data transformers convert raw cluster data in the relational database into a format usable for reporting and analysis.

Contents

- ◆ [About data transformers](#) on page 61
- ◆ [Configuration to modify data transformer behavior](#) on page 64
- ◆ [Data transformer actions](#) on page 64

About data transformers

The LSF and license data is logged in the relational database in a raw format. At regular intervals, the data transformer converts this data to a usable format.

Logging levels

There are logging levels that determine the detail of messages that the data transformers record in the log files. In decreasing level of detail, these levels are ALL (all messages), TRACE, DEBUG, INFO, WARN, ERROR, FATAL, and OFF (no messages).

By default, the data transformers log messages of INFO level or higher (that is, all INFO, WARN, ERROR, and FATAL messages).

The data transformer log files are located in the `datatransformer` subdirectory of your Analytics server log directory:

- ◆ UNIX (Solaris or Linux): `ANALYTICS_TOP/log/datatransformer`
- ◆ Windows: `ANALYTICS_TOP\log\datatransformer`

Default behavior

Data transformers convert data at regular intervals (hourly or daily). The following is a list of the data transformers and the database tables in which the data transformers generate the data:

About data transformers

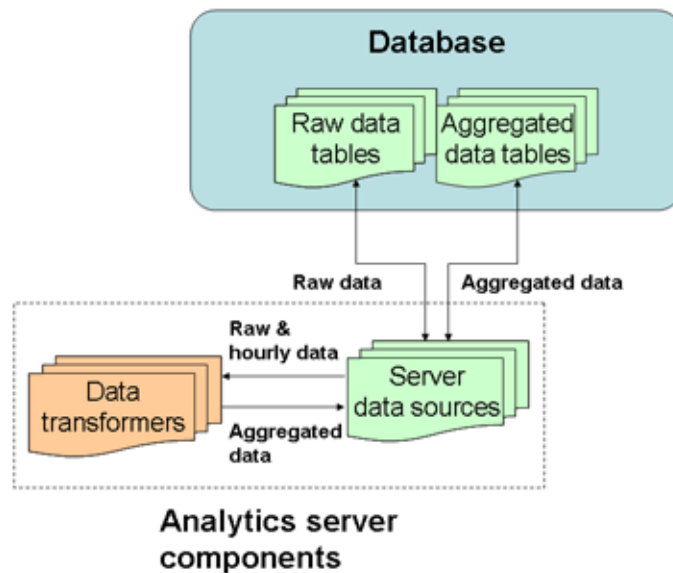
| Data transformer name | Transformed database tables |
|-----------------------------------|---|
| flexEventDaily | WI_FLEXEVENTMARTDAILY |
| flexEventHourly | WI_FLEXEVENTMART WI_LSFFLEXEVENTMART WI_LSFFLEXEVENTMARTHOLDING |
| hostGroupConfDaily | WI_HOSTGROUPCONFDAILY |
| hostGroupConfHourly | WI_HOSTGROUPCONF |
| hostMetricsDaily | WI_HOSTMETRICSDAILY |
| hostMetricsHourly | WI_HOSTMETRICS |
| hostPropertiesDaily | WI_HOST_BOOLEANRES WI_HOSTMETRICSDAILY WI_HOSTPROPERTIES |
| hostPropertiesHourly | WI_HOST_BOOLEANRES WI_HOSTMETRICS WI_HOSTPROPERTIES |
| licConsumpDaily | WI_LICENSECONSUMPTMARTDAILY |
| licDenialByFeatureDaily | WI_LICENSEDENIALBYFEATUREMARTDAILY |
| licDenialDaily | WI_LICENSEDENIALMARTDAILY |
| licDenialHourly | WI_LICENSEDENIALMART |
| licHourly | WI_LICENSECONSUMPTMART WI_LICENSETOTAL WI_LICENSEUTILMART WI_LICUSAGEBYUSERGRPMART |
| licUsageByUserDaily | WI_LICENSEUSAGEBYUSERMARTDAILY |
| licUsageByUsrgrpDaily | WI_LICBYUSERGRPMARTDAILY |
| licUtilByFeatureDaily | WI_LICUTILFEATUREMARTDAILY |
| licUtilDaily | WI_LICENSEUTILMARTDAILY |
| IsfDetailPendingReasonHourly | WI_DPR_BYINTERVAL WI_PENDJOBNUM_BYGROUPS WI_PENDTIME_BYGROUPS |
| IsfDetailPendingReasonJobNumDaily | WI_PENDJOBNUM_BYGROUPSDAILY |
| IsfDetailPendingReasonTimeDaily | WI_PENDTIME_BYGROUPSDAILY |
| IsfflexEventDaily | WI_LSFFLEXEVENTMART |
| IsfHostStatusHourly | WI_HOSTSTATUS |
| IsfJobExecProcHourly | WI_JOBEXECPROCS |
| IsfJobmartDaily | WI_JOB MARTDAILY |
| IsfJobmartHourly | WI_JOB MART WI_JOBRESUSAGECOST |
| IsfJobPendingReasonDaily | WI_JOBSPENDREASONDAILY |
| IsfJobPendingReasonHourly | WI_JOBSPENDREASON |
| IsfJobResUsageByGroupDaily | WI_JOBRESUSAGEBYGROUPDAILY WI_JOB SLOTUSAGEBYGROUPDAILY |

| Data transformer name | Transformed database tables |
|-----------------------------------|--|
| IsfJobResUsageByGroupHourly | WI_JOBRESUSAGEBYGROUP WI_JOB SLOTUSAGEBYGROUP |
| IsfJobResUsageCostDaily | WI_JOBRESUSAGECOSTDAILY |
| IsfJobStatisticsDaily | WI_NUMBEROFJOBSDAILY |
| IsfJobStatisticsHourly | WI_NUMBEROFJOBS |
| IsfJobThroughputDaily | WI_JOBTHROUGHPUTDAILY |
| IsfJobThroughputHourly | WI_JOBTHROUGHPUT |
| IsfSharedResourcesDaily | WI_SHAREDRESOURCEUSAGE DAILY |
| IsfSharedResourcePropertiesHourly | WI_RESOURCEPROPERTIES |
| IsfSharedResourceUsageHourly | WI_SHAREDRESOURCEUSAGE |
| userGroupConfHourly | WI_USERGROUPCONF |

Data transformer interactions

Data transformers obtain raw data and aggregated hourly data from the data tables through the server data sources, and stores aggregated data (including daily data) into the data tables through the server data sources.

The following diagram illustrates the interaction between the data transformers and other components.



Configuration to modify data transformer behavior

| Action | Configuration files | Parameter and syntax |
|---|--|---|
| Specify the default log level of your data transformer log files. | log4j.properties File location: ANALYTICS_TOP/conf | log4j.appender.\${datatransformer}=log_level, \${datatransformer} where ◆ <i>log_level</i> is the default log level of your data transformer log files. The data transformer only logs messages of the same or lower level of detail as <i>log_level</i> . Therefore, if you change the log level to <code>ERROR</code> , the data transformer will only log <code>ERROR</code> and <code>FATAL</code> messages. |

Data transformer actions

Data transformers are installed as scheduled tasks. To change the schedule of data transformers, see [Scheduled task actions](#) on page 79.

Events

An event is a change in Analytics reflecting a change in state.

Contents

- ◆ [About events](#) on page 65
- ◆ [Event actions](#) on page 65

About events

An event is a change in Analytics reflecting a change in state, including events that provide information about problems encountered when running Analytics (Warning, Error, or Fatal events), or events that contain useful administration information on Analytics activities (Info events).

Event notification

Analytics sends an event notification email when it encounters a change in state that matches the event notification settings. An event notification email informs the you of the change in state in Analytics or the cluster, allowing you to decide whether you want to check the Analytics console for further details.

Event actions

Actions on events

| Action | Analytics Console |
|-----------------------------------|--|
| View the list of events. | In the navigation tree, click Events . |
| View a filtered list of events. | When viewing the list of events, select Action > Filter Events from the menu toolbar. |
| Edit event notification settings. | When viewing the list of events, select Action > Notification from the menu toolbar. |
| | IMPORTANT: If you enable or disable event notification, you need to restart the Platform Task Scheduler to apply this change. See Restart the Platform Task Scheduler on page 66. |

Restart the Platform Task Scheduler

If you enable or disable event notification, you need to restart the Platform Task Scheduler to apply this change. The steps you take to restart the task scheduler depend on your operating system.

-
- ◆ Windows: Restart the task scheduler service.
 - a From the Windows Control Panel, select **Administrative Tools > Services**.
 - b Right-click **Platform Task Scheduler *version_number*** and select **Restart**.
 - ◆ UNIX (Solaris or Linux): Restart the task scheduler daemon.
 - a From the command line, navigate to the *ANALYTICS_TOP/bin* directory.
 - b Restart the Analytics daemons.

```
daemons.sh stop
daemons.sh start
```
-

Data purger

The data purger (`purger`) service maintains the size of the database by purging old data from the database.

Contents

- ◆ [About the data purger](#) on page 67
- ◆ [Data purger actions](#) on page 68

About the data purger

The relational database needs to be kept to a reasonable size to maintain optimal efficiency. The data purger manages the database size by purging old data from the database at regular intervals.

Logging levels

There are logging levels that determine the detail of messages that the data loaders record in the log files. In decreasing level of detail, these levels are ALL (all messages), TRACE, DEBUG, INFO, WARN, ERROR, FATAL, and OFF (no messages).

By default, the data purger logs messages of ERROR level or higher (that is, all ERROR and FATAL messages) to the data purger log file, which is located in the Analytics server log directory (`ANALYTICS_TOP/log` in the Analytics server host).

Default behavior

The data purger runs as two scheduled tasks on the Analytics server, but are disabled by default:

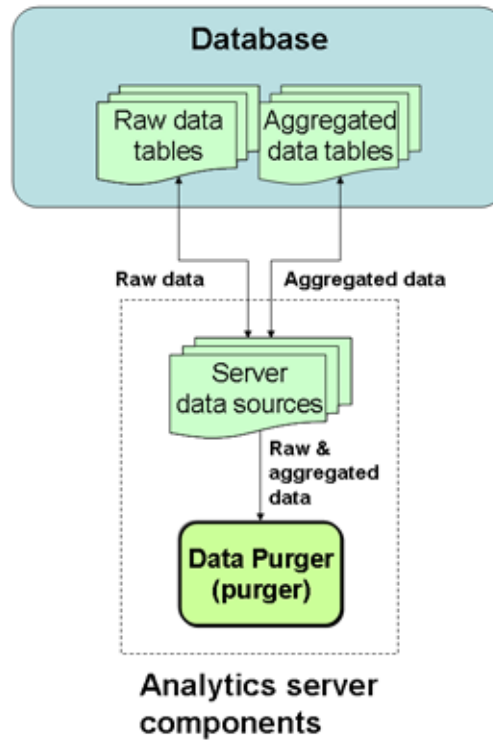
- PurgerRawData** Purges raw data from both the PERF and Analytics raw data tables. This data purger is disabled by default.
- PurgerAggData** Purges aggregated data from the aggregated data tables. This data purger is disabled by default.

Data purger interactions

When enabled, the data purger purges old records from the raw and aggregated data tables through the server data sources.

Data purger actions

The following diagram illustrates the interaction between the data purger and other components.



Data purger actions

The data purger is installed as a scheduled task. To change the schedule of the data purger service, see [Scheduled task actions](#) on page 79.

Datamarts

A datamart is a repository of data, gathered from operational data and other sources, that is used to build a cube. The datamart is generated by datamart ETL flows, which contains ETL streams that define how raw collected data is transformed into data ready for loading into cubes.

Contents

- ◆ [About datamarts](#) on page 69
- ◆ [Datamart actions](#) on page 72

About datamarts

Analytics uses datamarts to build cubes of the same names (with some exceptions, such as the "Host Properties" datamart). The datamarts contain ETL (Extract, Transform, and Load) streams defining how raw collected data is transformed into data ready for loading into cubes.

A datamart is a comma-separated file (.csv) created by the datamart ETL flow process that provides information required by Cognos PowerPlay Transformer to build a cube.

At cube build time, Analytics queries the database and writes the required data to the datamart file. During this process, Analytics can also add custom columns to the datamart by executing pre-defined calculations. When a cube is required, Analytics uses the `runETL` script to query the data source, applies any additional transformations, and writes the data to a datamart. Then, the `runBuild` script executes the Cognos commands to build the cube from the datamart and publish it online. Analytics rebuilds a datamart if there has been a change to any of the properties of the datamart file, or if the datamart has expired. By default, datamarts are always rebuilt.

The following table shows a list of datamarts and the cubes to which the datamarts contribute.

About datamarts

| Datamart | Cubes to which the datamart contributes |
|--|--|
| Daily FLEX license | Daily FLEX license |
| Hourly FLEX license | Hourly FLEX license |
| Daily hardware | Daily hardware |
| Hourly hardware | Hourly hardware |
| Daily hardware by host group | Daily hardware by host group |
| Hourly hardware by host group | Hourly hardware by host group |
| Hardware performance prediction | Hardware performance prediction |
| Daily host availability | Daily host availability |
| Hourly host availability | Hourly host availability |
| Host properties | Hardware performance prediction Daily host availability Hourly host availability Daily host status Hourly host status Hourly job throughput Hourly workload Hourly workload license |
| Daily host status | Daily host status |
| Hourly host status | Hourly host status |
| Daily job information | Daily workload |
| Hourly job information | Hourly workload |
| Hourly job reserved resource consumption | Hourly job resource consumption |
| Daily job reserved resource usage | Daily job resource consumption |
| Daily job slot usage by group | Daily job slot usage |
| Hourly job slot usage by group | Hourly job slot usage |
| Daily job statistics | Daily job statistics |
| Hourly job statistics | Hourly job statistics |
| Daily job throughput | Daily job throughput |
| Hourly job throughput | Hourly job throughput |
| Job throughput prediction | Job throughput prediction |
| Daily jobs by pending reason | Daily jobs by pending reason |
| Hourly jobs by pending reason | Hourly jobs by pending reason |
| Daily license consumption | Daily license consumption |
| Hourly license consumption | Hourly license consumption |
| Daily license denial | Daily license denial |
| Hourly license denial | Hourly license denial |
| Daily license denial by feature | Daily license denial by feature |
| Hourly license denial by feature | Hourly license denial by feature |
| Daily license usage by user | Daily license usage by user |

| Datamart | Cubes to which the datamart contributes |
|---------------------------------------|---|
| Hourly license usage by user | Hourly license usage by user |
| Daily license usage by user group | Daily license usage by user group |
| Hourly license usage by user group | Hourly license usage by user group |
| Daily license utilization | Daily license utilization |
| Hourly license utilization | Hourly license utilization |
| Daily license utilization by feature | Daily license utilization by feature |
| Hourly license utilization by feature | Hourly license utilization by feature |
| License utilization prediction | License utilization prediction |
| Daily shared resource usage | Daily shared resource |
| Hourly shared resource usage | Hourly shared resource |
| Daily workload license | Daily workload license |
| Hourly workload license | Hourly workload license |

Default behavior

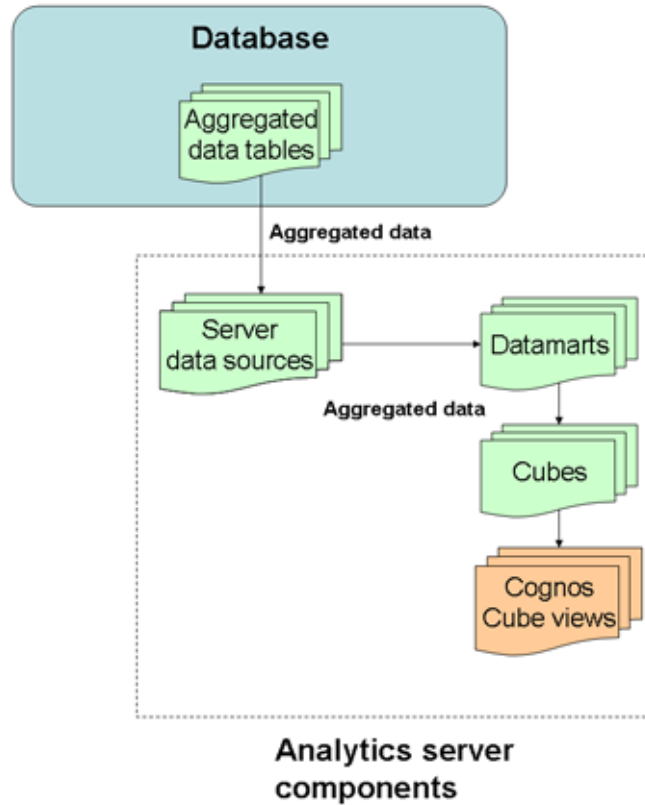
Analytics always rebuilds a datamart when a cube is built.

Datamart interactions

Datamart ETL flows obtain data from the aggregated data tables through the server data sources. Datamarts serve as a source of data for cubes. The data obtained by the cubes can be loaded in Cognos cube views.

Analytics collects raw data and stores it in raw data tables in the database to enable data processing such as aggregation and reformatting. The data is later extracted from the database and used to build datamarts, which are later used to build OLAP cubes.

The following diagram illustrates the interaction between datamarts and other components.



Datamart actions

Actions on the datamarts

| Action | Analytics Console |
|-------------------------------|---|
| View a list of the datamarts. | In the navigation tree, click Datamart ETL Flows . |

Edit or view a datamart

Edit or view a datamart ETL flow using the Datamart Properties window.

- 1 In the tree view of the Analytics Console, click **Datamart ETL Flows**.
- 2 Right-click the datamart in the main window and select **Edit Datamart**.
The **Datamart Properties** window for the datamart displays.
- 3 If you want to modify the duration of the datamart (that is, the data range of the data in the datamart), click the **ETL** tab and modify the **Date Selection Range** fields.

CAUTION: Do not make any other changes to the datamart properties. Doing so may result in problems with building cubes.

Cubes

A cube is a multidimensional data source that contains data measures organized into dimensions (data elements).

Contents

- ◆ [About cubes](#) on page 73
- ◆ [Cube actions](#) on page 76

About cubes

Cubes bring together key dimensions on your data. You can examine any dimensional combination of data. Refer to *Platform Analytics Cube Descriptions* for a list of the sample cubes included with Analytics.

Cognos PowerPlay Transformer uses a cube model file (.mdl) and one or more datamart files (.csv) to build cubes (.mdc). A datamart file lists required information to build a cube.

The following table shows a list of cubes and the datamarts that contributed to the cube build.

| Cube | Datamarts that contributed to the cube build |
|---------------------------------|--|
| Daily FLEX license | Daily FLEX license |
| Hourly FLEX license | Hourly FLEX license |
| Daily hardware | Daily hardware |
| Hourly hardware | Hourly hardware |
| Daily hardware by host group | Daily hardware by host group |
| Daily host availability | Daily host availability Host properties |
| Hourly host availability | Hourly host availability Host properties |
| Hourly hardware by host group | Hourly hardware by host group |
| Hardware performance prediction | Hardware performance prediction Host properties |

About cubes

| Cube | Datamarts that contributed to the cube build |
|---------------------------------------|--|
| Daily host status | Host properties Daily host status |
| Hourly host status | Host properties Hourly host status |
| Daily job resource consumption | Daily job reserved resource usage |
| Hourly job resource consumption | Hourly job reserved resource consumption |
| Daily job slot usage | Daily job slot usage by group |
| Hourly job slot usage | Hourly job slot usage by group |
| Daily job statistics | Daily job statistics |
| Hourly job statistics | Hourly job statistics |
| Daily job throughput | Daily job throughput |
| Hourly job throughput | Host properties Hourly job throughput |
| Job throughput prediction | Job throughput prediction |
| Daily jobs by pending reason | Daily jobs by pending reason |
| Hourly jobs by pending reason | Hourly jobs by pending reason |
| Daily license consumption | Daily license consumption |
| Hourly license consumption | Hourly license consumption |
| Daily license denial | Daily license denial |
| Hourly license denial | Hourly license denial |
| Daily license denial by feature | Daily license denial by feature |
| Hourly license denial by feature | Hourly license denial by feature |
| Daily license usage by user | Daily license usage by user |
| Hourly license usage by user | Hourly license usage by user |
| Daily license usage by user group | Daily license usage by user group |
| Hourly license usage by user group | Hourly license usage by user group |
| Daily license utilization | Daily license utilization |
| Hourly license utilization | Hourly license utilization |
| Daily license utilization by feature | Daily license utilization by feature |
| Hourly license utilization by feature | Hourly license utilization by feature |
| License utilization prediction | License utilization prediction |
| Daily shared resource | Daily shared resource usage |
| Hourly shared resource | Hourly shared resource usage |
| Daily workload | Daily job information |
| Hourly workload | Host properties Hourly job information |
| Daily workload license | Daily workload license |
| Hourly workload license | Host properties Hourly workload license |

Dimensions

Dimensions make up the structure of a cube. A dimension is a broad grouping of descriptive data about a major aspect of your workload, such as the number of CPUs, jobs, queues, projects, and hosts. Each cube has a different set of dimensions. The dimensions of the cube displays in the dimension line when the cube opens in the Cognos PowerPlay Web Explorer.

Category

Categories are different levels in each dimension. Categories are the items in the rows or columns of a Cognos cube view report. The intersection of all categories provide the values in your report.

Child category

Child categories are categories that are one level below another category.

Metric

Metrics are a specialized dimension of dynamic data that Analytics collects to determine host performance.

Measures

Measures are the data content of a cube. The actual values of measures are displayed in the crosstab (tabular) Cognos cube view report. A measure is a quantifiable statistic (amount) built into a cube. It is the actual data that Analytics collects and reports.

A measure may be a simple summary of available information, such as the total number of jobs, or a calculated value, such as the average memory used by a job. Measures can be displayed either as actual values or as percentages of aggregated values.

Cognos cube views

Cognos cube views are included in Analytics to help analyze the type of data included in the cube.

Default behavior

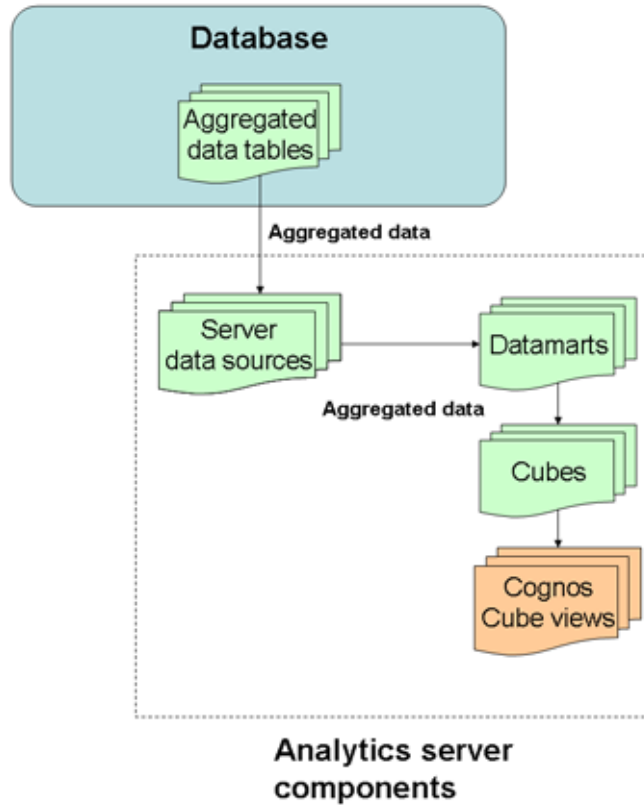
When you open a cube, the first and second dimensions are displayed respectively as rows and columns in the display area.

Cube interactions

Datamarts obtain data from the aggregated data tables through the server data sources. Datamarts serve as a source of data for cubes. The data obtained by the cubes can be loaded in Cognos cube views.

Cubes are a data source that obtain data from datamarts (which, in turn, obtain data from the aggregated data tables through the server data sources). The data obtained by the cubes can be loaded in Cognos cube views.

The following diagram illustrates the interaction between cubes and other components.



Cube actions

Actions to open the cubes and Cognos cube views

| Action | Cognos PowerPlay Web Explorer |
|-------------------------------------|--|
| Open a cube. | The cube name, beside a cube icon |
| Open a series of Cognos cube views. | The Cognos cube view series name, beside a NewsBox folder icon |
| Open a Cognos cube view. | The Cognos cube view series name, beside a NewsBox folder icon, then the Cognos cube view within the series, beside a Cognos cube view icon. |

Actions to manage cubes and Cognos cube views

See [Manage cubes](#) on page 99 for further details.

Scheduled tasks

Scheduled tasks are automated processing tasks that regularly run JavaScript-based scripts.

Contents

- ◆ [About scheduled tasks](#) on page 77
- ◆ [Scheduled task actions](#) on page 79

About scheduled tasks

After metric data is collected from hosts, the data undergoes several processing tasks on its way to being displayed in cubes and reports. Analytics automates the data processing by scheduling these processing tasks to run regularly. Each of these tasks calls a JavaScript-based script.

You can modify these tasks, reschedule them, and create new scheduled tasks.

Scripts

Analytics scheduled tasks call JavaScript-based scripts such as `runBuild`. These scripts read datamart definition, configuration, and scheduling files, then build and publish cubes.

Predefined scheduled tasks

Analytics includes several predefined scheduled tasks.

Build cubes (`BuildCubes`)

The `BuildCubes` scheduled task builds OLAP cubes. By default, this scheduled task builds all cubes (except the Daily Hardware by Host Group and Daily BLD License UsageDaily Distinct Users and the Weekday Performance cubes) in Analytics at 2:00 a.m. each day. The cubes are published to Cognos Upfront after they are successfully built.

About scheduled tasks

If a single scheduled task includes multiple cubes, then the cubes are built sequentially, which is the most efficient way of processing.

TIP: If you want to build some cubes at one interval and other cubes at another interval, you should create a new scheduled task for each interval. Multiple scheduled tasks can call the same JavaScript script and function to build cubes.

Building multiple cubes

You can create multiple scheduled tasks to build cubes. However, if scheduled tasks overlap (one scheduled task's cubes are not finished building when another schedule begins to build its cubes) you may notice performance problems.

You can include a cube in multiple scheduled tasks. If a cube build started by one task is still in progress when another task attempts to build the same cube, the second schedule task's build will log an error instead of starting.

You can build multiple cubes either sequentially or simultaneously (in parallel). With large amounts of data, Analytics builds cubes faster if they are built simultaneously. You should create an additional task that runs the `buildcubes.js` script, and then both tasks can run simultaneously. If one task tries to build a cubes when it is being built by another cubes, that task just skips building the cubes. However, this requires multiple processors and additional temp space.

Build datamarts (BuildDatamarts)

The BuildDatamarts scheduled task builds datamarts without building OLAP cubes.

NOTE: You cannot modify this scheduled task (or any scheduled task that builds datamarts) once it has already started to build datamarts. In this case, the dialog box opens in read-only mode.

Building multiple datamarts

You can build multiple datamarts either sequentially or simultaneously (in parallel). With large amounts of data, Analytics builds datamarts faster if they are built simultaneously. You should create an additional task that runs the `bulddatamarts.js` script, and then both tasks can run simultaneously. If one task tries to build a datamart when it is being built by another datamart, that task just skips building the datamart. However, this requires multiple processors and additional temp space.

Data purger (Purger*)

The purger scheduled tasks, which all have "Purger" in their names, control the data purger. For more information, see [Data purger](#) on page 67.

Data transformers (*DataAgg*)

The data transformer scheduled tasks, which all have "DataAgg" in their names, control the data transformers. For more information, see [Data transformers](#) on page 61.

Scheduled task actions

Actions on the scheduled tasks

| Action | Analytics Console |
|--|---|
| View a list of scheduled tasks. You need to do this to perform any other action on the scheduled tasks. | In the navigation tree, click Scheduled Tasks . |
| Create a task in the list of scheduled tasks. | See Create, edit, or view a scheduled task on page 79 for detailed information. |
| View or edit a task from the list of scheduled tasks. | See Create, edit, or view a scheduled task on page 79 for detailed information. |
| Remove a task from the list of scheduled tasks. | In the main window, right-click the scheduled task and select Remove Scheduled Task . |
| Run a task manually from the list of scheduled tasks. | In the main window, right-click the scheduled task and select Run Now . |

Create, edit, or view a scheduled task

Create, edit, or view a scheduled task.

You might edit a scheduled task for the following reasons:

- ◆ Schedule a task that is currently unscheduled
- ◆ Edit the next run time
- ◆ Edit the run interval
- ◆ Add or edit task parameters
- ◆ Modify how information about the task is logged and where it is stored
- ◆ Modify the JavaScript file and function called by the task

- 1 In the navigation tree of the Analytics Console, select **Scheduled Tasks**.
- 2 Select the scheduled task to create, edit, or view.
 - ❖ To create a new scheduled task, right-click on the main window and select **Add Scheduled Task**.
 - ❖ To edit or view an existing scheduled task, right-click the scheduled task in the main window and select **Edit Scheduled Task**.

The **Scheduled Task** window for the scheduled task is displayed.

For an existing scheduled task, the following information is displayed in addition to the scheduled task parameters:

- ◆ **Last Run Time:** The previous time that this scheduled task was run.
- ◆ **Last Run Status:** The status of the last run of this scheduled task.
- ◆ **Last Checkpoint:** The last time the data was checkpointed during the scheduled task. If the checkpoint and the scheduled task are completed, this is "DONE".

3 Edit the scheduled task parameters that you want to change.

CAUTION: Do not change the name of the scheduled task; otherwise, Analytics may have problems with scheduling your renamed task.

- a** To change the script file for the task, specify the new script file in the **Script File** field.

The script file must reside in the `ANALYTICS_TOP` directory. If it is in a subdirectory, include the file path of the subdirectory in the field.

For example, if the new script file is `new_script.js` and resides in the `ANALYTICS_TOP/bin` directory, define the new script file as the following:

```
/bin/new_script.js
```

- b** To change the function to run in the script for the task, specify the new script function in the **Script Function** field.

The script can include other functions, but the other functions will run only if they are called by this specified script function.

- c** To change the log file for this task, specify the new log file in the **Log File** field.

The location of the log directory is as follows:

- ◆ UNIX (Solaris or Linux): `ANALYTICS_TOP/log`
- ◆ Windows: `ANALYTICS_TOP\log`

- d** To change the level of detail of information recorded in the log file, select the new log level in the **Log Level** field.

All messages of this level or lower are recorded in the log file. In decreasing level of detail, the logging levels are `DEBUG`, `VERBOSE`, `INFO`, `WARNING`, and `ERROR`.

For example, if you specify "INFO", the log file contains `INFO`, `WARNING`, and `ERROR` messages.

- e** To enable scheduling for this task, enable the **Enable Scheduling** check box.
- f** To change the next date and time that this task is scheduled to run, modify the fields in the **Next Run Time** box.
- g** To change the run interval of the scheduled task to a fixed interval, select the **Run every:** field and specify the interval.
- h** To change the run interval of the scheduled task to a calculated value, select the **Call this function** field specify the function in the script file to determine the run interval.

The function must return a time stamp string in the following format:

```
YYYY-MM-DD hh:mm:ss.xxxx
```

This time stamp indicates the the next date and time in which this task is scheduled to run.

- i** To add optional parameters that Analytics looks for in the script file, enter them into the **Parameters** field.

This field does not exist in certain scheduled tasks.

- 4 For the `BuildCubes` scheduled task, select additional build options.
 - a To specify the cubes to build, enable the check box of each cube that you want to build in the **Cubes** field. Click **All** to toggle between selecting and clearing the check boxes for all cubes.
 - b To publish the cubes after they are built, enable the **Publish** check box. Otherwise, clear this box.
 - 5 For the `BuildDatamarts` scheduled task, select the datamarts you want to build.

To specify the datamarts to build, enable the check box of each datamart that you want to build in the **Datamarts** field. Click **All** to toggle between selecting and clearing the check boxes for all datamarts.
 - 6 To save your changes and close the window, click **OK**.
-

Scheduled task actions

Analytics server command-line tools

Contents

- ◆ [daemons](#) on page 85
- ◆ [runBuilder](#) on page 87
- ◆ [runconsole](#) on page 89

Contents

daemons

Stops and restarts Analytics on the server.

Synopsis

```
daemons.sh start | stop
```

Options

start

Starts the Analytics daemons on the server.

stop

Stops the Analytics on the server.

If you do not specify a cube name, `runBuilder` builds all cubes in the Analytics product. To build a specific cube, you must also specify the package name.

Options

runBuilder

Builds cubes in Analytics. Also builds the required datamarts if they have expired. Optionally publishes cubes to Cognos Upfront.

Synopsis

```
runBuilder.sh [package_name] [cube_name] [-p]
```

```
runBuilder [package_name] [cube_name] [-p]
```

`runBuilder.sh` is the command for UNIX (Solaris or Linux) and `runBuilder` is the command for Windows.

Description

If you execute `runBuilder` with no parameters, `runBuilder` checks the `ANALYTICS_TOP/conf/schedules.xml` file to see if any packages or cubes are scheduled to build, and then builds them.

If you specify a package name, `runBuilder` checks the `ANALYTICS_TOP/conf/packages.xml` file for a list of all the registered Analytics products, and the location of their package definition files. Then it checks each package definition file for the package name. If it finds the specified package name, `runBuilder` builds the package.

If you also specify a cube name, `runBuilder` checks the package definition file for a list of all the cubes in the package, and the locations of the cube definition files. Then it checks each cube definition file for the cube name. If it finds the specified cube name, `runBuilder` builds the cube.

`runBuilder` checks the expiry date of the required datamarts and builds them using `runETL` if necessary.

`runBuilder` checks the `ANALYTICS_TOP/conf/config.xml` file to get the Cognos bin directory and administrator password. `runBuilder` uses this information to publish cubes. Enter this information in the console, or by editing the `config.xml` file directly.

`runBuilder` writes a separate log file for each cube it builds in the directory where the cube's model file (`.mdl`) is located.

Options

package_name

Optional. The name of an Analytics product. This is the name displayed in the `<Name></Name>` tags of the package definition file.

cube_name

Optional. The name of a cube in that Analytics product. This is the name displayed in the **Cube Properties** dialog box of the console or the within the `<Name></Name>` tags of the cube definition file.

If you do not specify a cube name, `runBuilder` builds all cubes in the Analytics product. To build a specific cube, you must also specify the package name.

Examples

- p Optional. Publishes to Cognos Upfront after a successful build. If not specified, `runBuilder` stops after the build is complete.

Examples

```
runBuilder
```

Checks the `ANALYTICS_TOP/conf/schedules.xml` file to see if any packages or cubes are scheduled to be built, and builds them.

```
runBuilder Analytics
```

Builds all cubes in Analytics, but does not publish them to Cognos Upfront.

```
runBuilder Analytics Hourly Hardware -p
```

Builds and publishes the Hourly Hardware cube in Analytics.

runconsole

Starts the Analytics console.

Synopsis

```
runconsole.sh
```

```
runconsole
```

`runconsole.sh` is the command for UNIX (Solaris or Linux) and `runconsole` is the command for Windows.

If you are running this command locally from the Analytics server running UNIX (Solaris or Linux), you need to be running X-Windows. If you are running this command remotely, you need to set your display environment.

Synopsis



Managing Analytics

Secure your data and working environment

Customize the security of your cluster to secure your data and working environment.

- ◆ [Add user classes and users to Cognos \(Windows\)](#) on page 93.
- ◆ [Add user classes to Cognos \(Solaris or Linux\)](#) on page 95.
- ◆ [Add a new user directly from Upfront](#) on page 96.
- ◆ [Open ports to communicate across firewalls](#) on page 96.
- ◆ [Modify the database password](#) on page 97.
- ◆ [Modify the Cognos administrator password](#) on page 97.

Add user classes and users to Cognos (Windows)

Control access to your Cognos Upfront pages and Cognos applications by creating user classes and users that do not have the same access privileges as the administrator.

The administrator user class is the *root user class*. Create new users that do not have the same access privileges by assigning membership of new users to various user classes with select access privileges.

- 1 Log into the server as the Cognos Administrator.
- 2 Launch the *Cognos Access Manager - Administration* tool.
Click **Start > Programs > Cognos Series 7 Version 4 > Tools > Access Manager - Administration**.
The **Access Manager - Administration** window displays.
- 3 In the left pane, click the **Directory Servers** folder.
- 4 Select **Actions > Add Connection**.
The **Directory Server Properties** window displays.
- 5 In the **General** tab, add a connection to the Directory Server host.

Add user classes and users to Cognos (Windows)

- a In the **Host** field, specify the fully-qualified name of the Directory Server host.
 - b In the **Port/SSL Port** field, specify the port used by the data directory server.

This port should be the same as the port you used earlier when creating the data directory server.

The port used by the data directory server is 389 by default. If you installed Cognos using the silent installation script, it uses the default port number.
 - c Click **Test** to verify your connection.
- 6 Add a new user class to the **Directory Servers** folder.
- a Navigate to the **Directory Servers > connection_name > Default** folder.
 - b Right-click **Root User Class** and select **Add User Class**.

The **User Class <New User Class> Properties** window displays.
 - c In the **General** tab, specify the name of the user class and assign access times for members of this user class.
 - d In the **Permissions** tab, assign the appropriate permissions for members of this user class.
 - ◆ Access permissions on Cognos servers, applications, and other user classes or users.
 - ◆ Permissions for members of this user class to modify its own properties.
 - e Click **Apply**, then **OK** to save the new user class to Cognos.

View the new user class in the tree on the left pane under **Directory Services > connection_name > Default > Root User Class**.
- 7 Add a new user to the **Directory Servers** folder.

TIP: You can also add a new user directly from Cognos Upfront instead of adding it to the **Directory Servers** folder.

- a Navigate to the **Directory Servers > connection_name > Default** folder.
- b Right-click **Users** and select **Add User**.

The **User <New User> Properties** window displays.
- c In the **General** tab, specify a name for the user.
- d In the **User Signons** tab, select **Basic Signon**.
- e In the **Basic Signon** dialog, specify the user ID, password, and password modification rules for the user.
- f In the **Memberships** tab, expand the **Root User Class** to view all the user classes and select the appropriate User Class for the user.
- g Click **Apply**, then **OK** to save the new user to Cognos.
 - ◆ View the new user class in the tree on the left pane under **Directory Services > connection_name > Default > Users**.

- ◆ Test the access privileges of the new user by logging on to Upfront.

Add user classes to Cognos (Solaris or Linux)

Control access to your Cognos Upfront pages and Cognos applications by creating user classes and users that do not have the same access privileges as the administrator.

The administrator user class is the *root user class*. Create new users that do not have the same access privileges by assigning membership of new users to various user classes with select access privileges.

- 1 Log into the server as the Cognos Administrator.

- 2 Connect to the root user account.

```
su root
```

- 3 Navigate to the Cognos cer5 binary directory.

```
cd /usr/cognos/cer5/bin
```

- 4 Create a new file with appropriate access permissions for a new user class.

For example, create a new file `userclass.txt` with content similar to the following to create a new user class named Platform User Class:

```
//LogonNamespace, default, Administrator
//Set the type of signon.
SetNamespaceProperty, "default", "ActiveSignon", "Both"
AddUserOSSignon, "Administrator", "Administrator"
AddUserOSSignon, "Administrator", "%%CurrentUser%%"
//Add User class
AddUserClass, "Platform User Class", "Root User Class"
SetUserClassProperty, "Platform User Class", "DataSourcePrivileges", "Show"
SetUserClassProperty, "Platform User Class", "PowerPlayServerPrivileges", "Show"
SetUserClassProperty, "Platform User Class", "TransformerServerPrivileges", "Show"
SetUserClassProperty, "Platform User Class", "UserClassPrivileges", "Show"
SetUserClassProperty, "Platform User Class", "UserPrivileges", "ModifySelf"
```

- 5 Open the Cognos Access Manager to add the new user class to Cognos.

For example, if the new user class information is saved in

```
/usr/cognos/cer5/bin/userclass.txt,
```

```
./AccessAdmMaint -uid=Administrator -pass=myspasswd
-filename=/usr/cognos/cer5/bin/userclass.txt -filetype=2
```

The log file is saved in the `/usr/cognos/cer5/bin` directory.

Postrequisites: Add new users directly from Upfront, then add the user to the user class.

Add a new user directly from Upfront

Use Upfront to create a new user and add the user to an existing user class.

- 1 Launch any web browser and visit the Cognos page in your Upfront gateway server:

`http://Analytics_server_host/cognos`

TIP: The Upfront gateway was installed with the PowerPlay Enterprise Server.

- 2 At the prompt, enter a user ID and password that is valid for your authentication source.

NOTE: If your user ID is not a member of the root user class, it must have access to the privileges to create users for this user class.

For example, if you are using the default namespace created when you tested the directory server, enter "Administrator" in the **User ID** field and leave the **Password** field blank.

Upfront displays in your web browser.

- 3 In the **NewsIndex** frame, click **Administration**.

- 4 Click **User Manager - add**.

The Add a New User form displays.

- 5 Complete the fields in the Add a New User form.

- a At the Enter the user information page, complete the new user information and password modification rules.
- b At the Select one or more user classes page, select the desired user classes for the new user.
- c Optional. Navigate the other pages to select optional properties for the user.
- d Click **Finish** to save the new user to Cognos.

Open ports to communicate across firewalls

If your cluster extends across the Internet securely, the server has to communicate with other hosts in the cluster across firewalls. Analytics uses the following ports to communicate with other hosts in the cluster:

| Port name | Default port number | Additional information |
|-----------|---------------------|---|
| PIAD_PORT | 9090 | Internal port for the task scheduler. Used for communicating with the automation manager. Configuration is not required. |
| PIEM_PORT | 9091 | Internal port for the event manager. Used for receiving events from Analytics components. Configuration is not required, |
| PIAM_PORT | 9092 | Internal port for the automation manager. Used for receiving events from Analytics components. Configuration is not required. |

- 1 Edit `ANALYTICS_TOP/conf/pi.conf` to open the appropriate ports.
- 2 Restart the Analytics Console to start communicating with the new ports.

Modify the database password

If you modify the password that Analytics data sources use to connect to the database, you must update Analytics to use the new password.

- 1 Log into the Analytics Console.
- 2 In the navigation tree, select **Data Sources**.
- 3 In the right pane, right-click **ReportDB** and select **Edit Data Source**.
The **Data Source Properties** window displays.
- 4 Specify the new password.
- 5 To verify the database connection, click **Test**.
- 6 To save your changes, click **OK**.

Modify the Cognos administrator password

If you modify the Cognos administrator password, you must update Analytics to use the new password.

- 1 Log into the Analytics Console.
- 2 Select **View > Options**.
The **Cognos** tab of the **Options** dialog box displays.
- 3 Specify the new password.
- 4 To save your changes, click **OK**.

Modify the Cognos administrator password

Manage cubes

Manage cubes and Cognos cube views.

- ◆ [Build a cube manually](#) on page 99.
- ◆ [Save a Cognos cube view](#) on page 100.
- ◆ [Export a Cognos cube view to a PDF file](#) on page 100.
- ◆ [Create a bookmark to a Cognos cube view](#) on page 100.
- ◆ [Modify cube properties](#) on page 101.

Build a cube manually

Prerequisites: Platform Analytics has been running for at least one day. Cube builds never include data from the current day.

Optional. By default, cube builds occur daily as a `Build Cubes` scheduled task. You need to build a cube manually to obtain a cube build outside of the schedule.

NOTE: You cannot build the cubes if you installed the Analytics console on a Linux host or on a host without Cognos, but you can still see the cubes.

- 1 In the Analytics Console, select **View > Options**.
- 2 In the **Cognos** tab, enable **Prompt to publish a newly built cube**.
- 3 In the navigation tree, select **Cubes**.
- 4 In the main window, right-click the cube to build and select **Build Cube**.
The **Build Cube** prompt appears.
- 5 To publish the cube after a successful build, select **Yes**.

Analytics launches a build log, which you can read while the cube is building. This log shows all the datamart building, cube building, cube publishing, and error information, and also states whether the cube build is successful.

Save a Cognos cube view

You can also retrieve the error information from `/logs/BuildCubes.log` in the installation directory.

- 6 When the cube build is complete, click **Close**.

Cube builds include relevant data from the previous day and earlier.

Save a Cognos cube view

Save a Cognos cube view to Cognos Upfront, the web interface, as a primary NewsItem in an Upfront NewsBox.

- 1 At a web browser window, open and edit the Cognos cube view that you want to save.
- 2 On the **Cognos PowerPlay Web Explorer** toolbar, click **Save As**, then specify the name, description, and destination location of your Cognos cube view.

TIP: In Cognos Upfront, other users who have access to the NewsBox can delete primary NewsItems in that NewsBox. If you do not want other users to delete the Cognos cube view, publish the cube view as a NewsItem in a personal NewsBox, then created a linked NewsItem by copying the NewsItem to the non-personal NewsBox.

Export a Cognos cube view to a PDF file

- 1 At a web browser window, open and edit the Cognos cube view that you want to export.
- 2 On the **Cognos PowerPlay Web Explorer** toolbar, click **Export PDF**.
The PDF version of the Cognos cube view displays in an Adobe Acrobat window.
- 3 To save the PDF file, press **Ctrl-S** and specify a name and location for your file.

Create a bookmark to a Cognos cube view

Create a URL to a specific Cognos cube view to return to the cube view at any time. You can save the resulting URL as a convenient bookmark in your web browser or a shortcut in your operating system desktop.

- 1 At a web browser window, open and edit the Cognos cube view that you want to save.
- 2 On the **Cognos PowerPlay Web Explorer** toolbar, click **Prepare Bookmark**.
The complete URL for the Cognos cube view appears in the location or address bar of your web browser.

The URL includes information about the categories, data formats, and filtering in the Cognos cube view.

Modify cube properties

You can modify the cube description or the directory location for the built cube file, but you should not modify other cube properties unless you are modifying a custom cube.

- 1 In the navigation tree of the Analytics console, select **Cubes**.
- 2 In the main window, right-click the cube to modify and select **Edit Cube**.
The **Cube Properties** dialog displays.
- 3 Modify any data source properties as necessary.

NOTE: You should only modify the default values for name, model file, or datamarts for custom cubes.

- a For custom cubes, in the **Name** field, change the name of the cube to be displayed in the Analytics console.
This change does not affect the name displayed in Cognos Upfront.
 - b In the **Description** field, change the description to be displayed in the Analytics console.
This change does not affect the description displayed in Cognos Upfront.
 - c In the **Cube File** field, change the directory location where Analytics builds the cube and stores the output cube file (.mdc).
 - d For custom cubes, in the **Model File** field, change the directory location of the Cognos Transformer cube model file (.mdl).
 - e For custom cubes, in the **Datamarts** section, click **Add** or **Remove** to manage the datamarts in the cube source.
Each cube must have at least one datamart, so you cannot remove the last datamart from the cube source.
To add datamarts to a cube or to create your own cubes, you must also edit the cube model files in Cognos PowerPlay Transformer.
- 4 Click **OK**.
-

Modify cube properties

Troubleshooting the server

Contents

- ◆ [Check the health of the Analytics server](#) on page 103
- ◆ [Check the Analytics server log files](#) on page 104
- ◆ [Check the status of the Analytics server database connection](#) on page 105

Check the health of the Analytics server

Use the Analytics Console to verify that the Analytics server is running correctly.

- 1 Log into the Analytics server host.
- 2 Launch the Analytics Console.
 - ❖ UNIX (Solaris or Linux): `ANALYTICS_TOP/bin/runconsole.sh`
 - ❖ Windows: **Start > Programs > Platform Analytics version_number Server > Platform Analytics version_number Console**
- 3 Click **Data Collection Node** in the navigation tree and verify that the node is running correctly.

To view the data loader properties, right-click each loader controller instance and select **Loader Properties**.
- 4 Click **Purger** in the navigation tree and compare the data purger settings with your cluster data retention policies.

To view the data purger settings, right-click each purger task and select **View Purger Properties**.
- 5 Click **Scheduled Tasks** in the navigation tree and verify that the scheduled tasks are running correctly according to schedule.
- 6 Click **Events** in the navigation tree and verify that there are no `ERROR` or `FATAL` events.
- 7 Verify the email notification settings.

While in **Events**, click **Action > Notification** to open the **Event Notification** dialog.

Check the Analytics server log files

- 8 Click **Cubes** in the navigation tree and check the status of your cubes.
 - a For any cubes with a "failed" status, right-click the cube and select **Build Cube** to manually build the cube.
 - b If the manual cube build still failed, click **Events** in the navigation tree and look for events related to your cube.

To find events specific to your cube, right-click the main window, select **Filter Events**, and enter the name of your cube in the **Keyword** field.
- 9 Click **Datamart ETL Flows** in the navigation tree and check the status of your datamarts.
 - a For any datamarts with a "failed" status, right-click the datamart and select **Build Datamart** to manually build the datamart.
 - b If the manual cube build still failed, click **Events** in the navigation tree and look for events related to your datamart.

To find events specific to your datamart, right-click the main window, select **Filter Events**, and enter the name of your datamart in the **Keyword** field.

Check the Analytics server log files

Check the Analytics server log files to verify that there are no errors.

- 1 Verify that there are no errors in the data transformer log files.

The data transformer log files (*data_transformer_name.log.host_name*) are located in the `datatransformer` subdirectory of the Analytics server log directory:

 - ◆ UNIX (Solaris or Linux): `ANALYTICS_TOP/log/datatransformer`
 - ◆ Windows: `ANALYTICS_TOP\log\datatransformer`
- 2 Verify that there are no errors in the scheduled cube build log files.

The scheduled cube build log files (*cube_name.log.host_name*) are located in the `cube` subdirectory of the Analytics server log directory:

 - ◆ UNIX (Solaris or Linux): `ANALYTICS_TOP/log/cube`
 - ◆ Windows: `ANALYTICS_TOP\log\cube`
- 3 Verify that there are no errors in the manual cube build log files.

The manual cube build log files (*cube_name.log.host_name*) are located in the `cube_manual_build` subdirectory of the Analytics server log directory:

 - ◆ UNIX (Solaris or Linux): `ANALYTICS_TOP/log/cube_manual_build`
 - ◆ Windows: `ANALYTICS_TOP\log\cube_manual_build`
- 4 Verify that there are no errors in the scheduled datamart build log files.

The scheduled datamart build log files (*datamart_name.log.host_name*) are located in the `datamart` subdirectory of the Analytics server log directory:

 - ◆ UNIX (Solaris or Linux): `ANALYTICS_TOP/log/datamart`
 - ◆ Windows: `ANALYTICS_TOP\log\datamart`

- 5 Verify that there are no errors in the manual datamart build log files.
The manual datamart build log files (*datamart_name.log.host_name*) are located in the *datamart_manual_build* subdirectory of the Analytics server log directory:
 - ◆ UNIX (Solaris or Linux): *ANALYTICS_TOP/log/datamart_manual_build*
 - ◆ Windows: *ANALYTICS_TOP\log\datamart_manual_build*
- 6 Verify that there are no errors in the data purger log file.
The data purger log file (*purger.log.host_name*) is located in the Analytics server log directory:
 - ◆ UNIX (Solaris or Linux): *ANALYTICS_TOP/log*
 - ◆ Windows: *ANALYTICS_TOP\log*
- 7 Verify that there are no errors in the event manager log file.
The event manager log file (*eventmanager.log.host_name*) is located in the Analytics server log directory:
 - ◆ UNIX (Solaris or Linux): *ANALYTICS_TOP/log*
 - ◆ Windows: *ANALYTICS_TOP\log*
- 8 Verify that there are no errors in the automation manager log file.
The automation manager log file (*automationmanager.log.host_name*) is located in the Analytics server log directory:
 - ◆ UNIX (Solaris or Linux): *ANALYTICS_TOP/log*
 - ◆ Windows: *ANALYTICS_TOP\log*

Check the status of the Analytics server database connection

Use the Analytics Console to verify the Analytics server database connection.

- 1 Log into the Analytics server host.
- 2 Launch the Analytics Console.
 - ❖ UNIX (Solaris or Linux): *ANALYTICS_TOP/bin/runconsole.sh*
 - ❖ Windows: **Start > Programs > Platform Analytics version_number Server > Platform Analytics version_number Console**
- 3 Click **Data Sources** in the navigation tree.
- 4 For each database entry in the main window, test the database connection.
 - a Right-click the database name and select **Edit Data Source**.
The **Data Source Properties** window displays.
 - b Click **Test** to test the database connection.

Check the status of the Analytics server database connection

- A**
- Analytics Console 59
- ANALYTICS_TOP 11
- architecture 11, 13
- B**
- BuildCubes 77
- BuildDatamarts 78
- C**
- category 75
 - child 75
- Cognos
 - administrator password 97
 - cube views 75
 - bookmark 100
 - exporting as PDF 100
 - saving 100
 - Upfont
 - users 96
 - user classes 93, 95
 - users 93, 95, 96
- cube views 75
 - bookmark 100
 - exporting as PDF 100
 - managing 99
 - saving 100
- cubes 73
 - Analytics Console 59
 - building manually 99
 - dimensions 75
 - category 75
 - child category 75
 - metric 75
 - managing 99
 - measures 75
 - properties 101
 - views 75
- D**
- daemons 85
- data loaders 31
 - data loss protection 34
 - FLEXIm 32
 - FLEXnet 32
 - frequency 34
 - license 32
 - log files 53
 - log levels 31, 61
 - default 34
 - temporary 35
 - LSF 32
 - LSF advanced data 32
 - scheduling 27
 - status 55
- data loss protection 34
- data mapping 59
- data purger 67
 - Analytics Console 59
 - log levels 67
- data sources 21
 - Analytics Console 59
 - node 22
 - status 56
 - password 97
 - ReportDB 21
 - server 21
 - status 105
- data transformers 61
 - log levels 64
- database 19
 - JDBC connection 21
 - password 97
 - ports 12
- datamarts 69
 - Analytics Console 59
 - editing 72
 - viewing 72
- dbconfig 39
- dimensions 75
 - category 75
 - child category 75
 - metric 75
- directories 11
- E**
- events 65
 - Analytics Console 59
 - filtering 65
 - notification 65
- F**
- files
 - license 15
- firewalls
 - ports 96
- FLEXIm
 - data loaders 32
- FLEXnet
 - data loaders 32
- J**
- JDBC connection 21
- L**
- LicDataAggDaily 78
- LicDataAggHourly 78
- license 15
 - data loaders 32
 - FLEXIm
 - data loaders 32
 - FLEXnet
 - data loaders 32

Index

- loader controller 27
 - data collection
 - disabling 54
 - log levels 27
 - default 28
 - temporary 29
 - status 34
- log files 53
- log levels
 - data loaders 31, 61
 - default 34
 - temporary 35
 - data purger 67
 - data transformers 64
 - default 53
 - loader controller 27
 - default 28
 - temporary 29
- LSF
 - data loaders 32
 - advanced data 32
- LSF_TOP 11
- LSFDataAggDaily 78
- LSFDataAggHourly 78
- M**
- measures 75
- metric 75
- N**
- node 25
 - Analytics Console 59
 - architecture 13
 - commands 37
 - dbconfig 39
 - perfadmin 41
 - plclient 43
 - files 45
 - perf.conf 47
 - ports 12
 - troubleshooting 53
- P**
- PERF 11
 - directories 11
 - services
 - loader controller 27
- perf.conf 47
- PERF_CONFDIR 11
- PERF_LOGDIR 11
- PERF_TOP 11
- PERF_WORKDIR 11
- perfadmin 41
- plclient 43
- ports 11, 96
- purger 67
 - Analytics Console 59
 - log levels 67
 - PurgerAggData 67, 78
 - PurgerRawData 67, 78
- R**
- ReportDB 21
- runBuilder 87
- runconsole 89
- S**
- scheduled tasks 59, 77
 - BuildCubes 77
 - BuildDatamarts 78
 - data purger 67
 - data transformers 61
 - LicDataAggDaily 78
 - LicDataAggHourly 78
 - LSFDataAggDaily 78
 - LSFDataAggHourly 78
 - PurgerAggData 67, 78
 - PurgerRawData 67, 78
 - scripts 77
- security 93
- server 57
 - Analytics Console 59
 - architecture 13
 - checking 103
 - commands 83
 - daemons 85
 - runBuilder 87
 - runconsole 89
 - log files 104
 - ports 11
 - troubleshooting 103
- services
 - loader controller 27
 - log files 53
- status
 - data loaders 55
 - data sources
 - node 56
 - server 105
 - loader controller 34
- system architecture 13
- T**
- troubleshooting
 - node 53
 - server 103
- U**
- user classes 93, 95
- users 93, 95, 96