
Administering Platform Analytics

Platform Analytics
Version 7.6
February 2010



Contents

	Part I: About Analytics ... 5
1	Introduction to Analytics 7
2	Architecture overview 9
3	Licensing 13
	Part II: Database host ... 15
4	Database 17
5	Data sources 19
	Part III: Analytics node ... 23
6	Loader controller 25
7	Data loaders 29
8	Analytics node command-line tools 39
9	Analytics node configuration files 43
10	Troubleshooting the node 49
	Part IV: Analytics server ... 53
11	Analytics Console 55
12	Data transformers 57
13	Event notification 63
14	Data purger 67
15	Datamarts 71
16	Cubes 79
17	Scheduled tasks 87
18	Analytics server command-line tools 93
19	Analytics server configuration files 99
	Part V: Managing Analytics ... 109
20	Secure your data and working environment 111

21	Manage cubes	117
22	Troubleshooting the server	121

P A R T



About Analytics

Introduction to Analytics

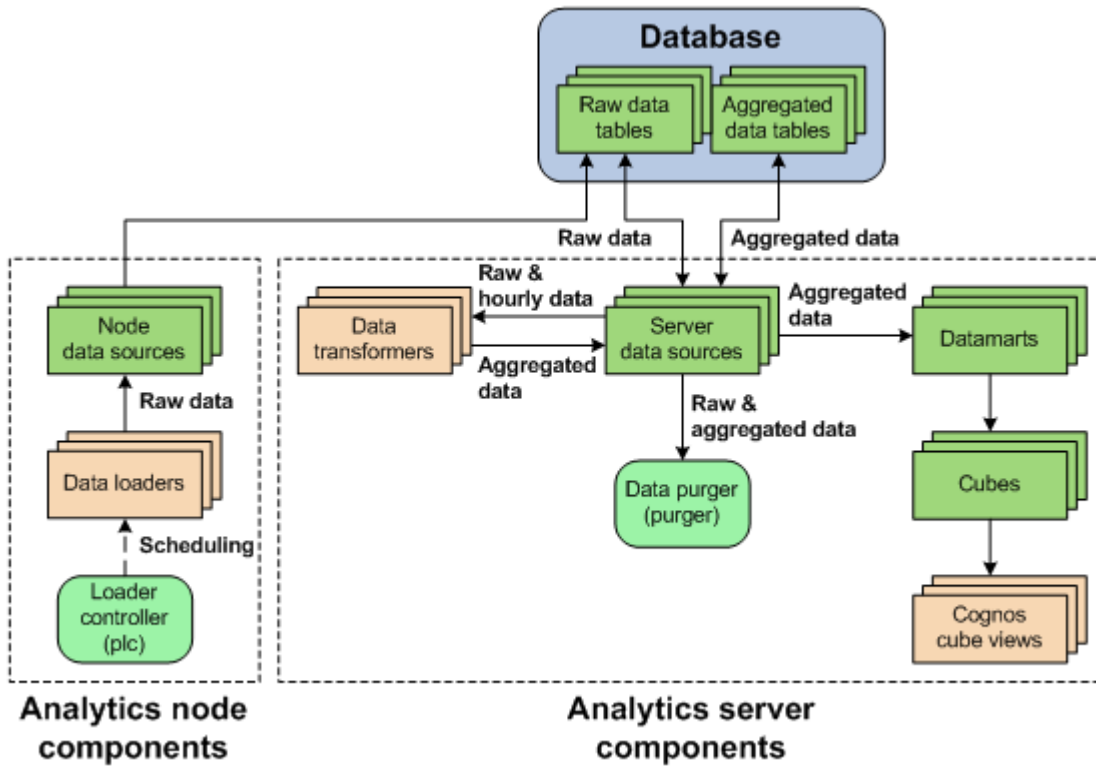
Platform Analytics is a workload intelligence solution for LSF cluster, FLEXnet license, and FLEXnet Manager license data. Analytics collects LSF and license data, aggregates it, and transforms it into data cubes for your analysis. Data cubes provide different views and correlations of your data. Analytics 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.

System architecture

The following diagram illustrates an overview of Analytics architecture.



System ports

For a list of ports that the Analytics hosts use, refer to *Installing Platform Analytics* (specifically, the *System ports* section in the *Analytics hosts* chapter).

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.

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>

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>

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*).

P A R T



Database host

Database host

4

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, 10g, and 11g 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.

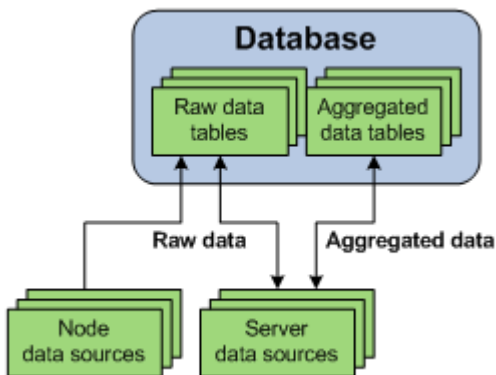
Depending on how you installed Analytics and set up the data schema, the database and its data schema may be partitioned. A partitioned database has tables divided into multiple, smaller tables. This improves database performance for larger clusters.

In a large database, purging old job records, transforming data, and other database maintenance tasks can have a significant effect on database performance. Purging old job records and transforming data from smaller tables has less of an impact on the system performance of active tables than on larger 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.



Data sources

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

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.

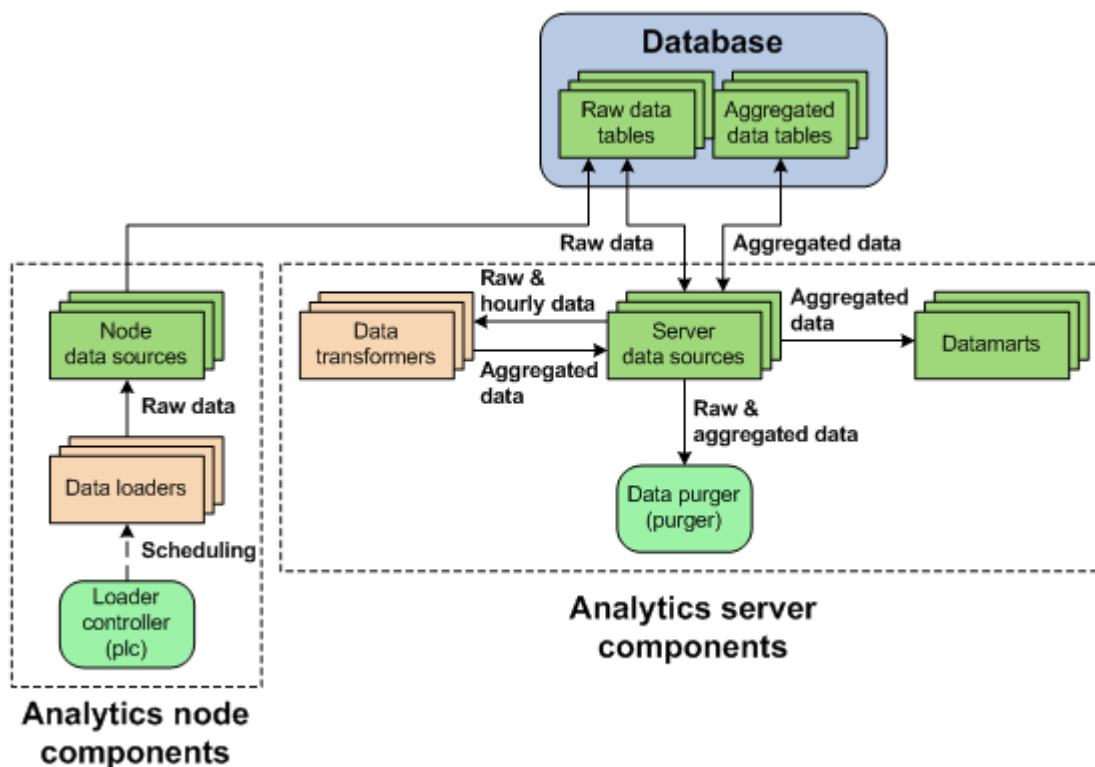
Datamarts obtain data from the aggregated data tables through the server data sources.

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

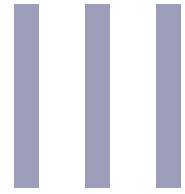
Actions on the Analytics server data sources

If the Analytics server is running on a UNIX 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: dbconfig.sh add <i>data_source_name</i> Windows: dbconfig add <i>data_source_name</i> where <ul style="list-style-type: none"><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: dbconfig.sh Windows: dbconfig
Edit the settings of any node data source, including FLEXnet Manager data sources.	UNIX: dbconfig.sh edit <i>data_source_name</i> Windows: dbconfig edit <i>data_source_name</i> where <ul style="list-style-type: none"><i>data_source_name</i> is the name the data source that you want to edit.



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.

Analytics node

Loader controller

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

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: \$PERF_LOGDIR
- Windows: %PERF_LOGDIR%

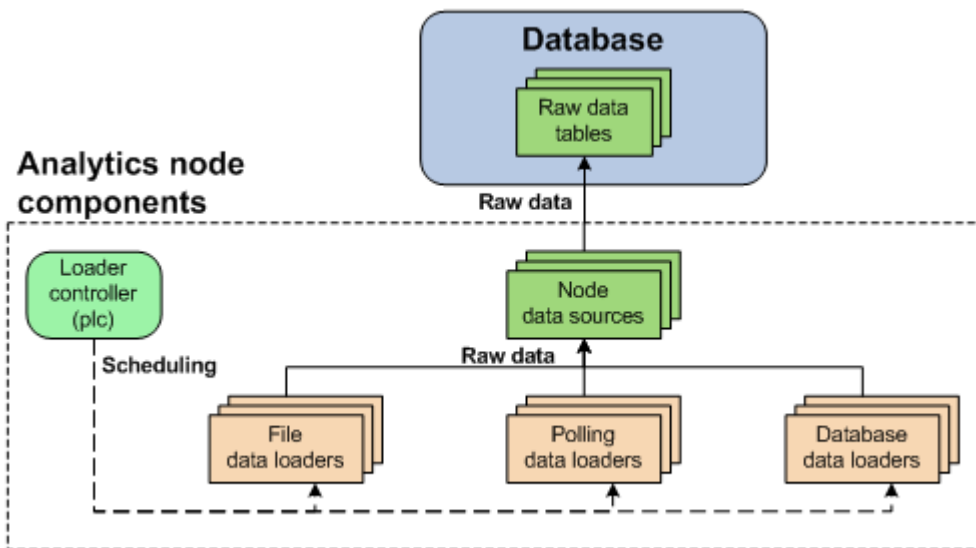
Default behavior

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 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 <code>plc</code> log file.	<code>log4j.properties</code> File location: UNIX: <code>\$PERF_CONFDIR</code> Windows: <code>%PERF_CONFDIR%</code>	<code>log4j.logger.com.platform.perf.data.loader=log_level, com.platform.perf.data.loader</code> where <ul style="list-style-type: none"> <code>log_level</code> 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 <code>log_level</code> . Therefore, if you change the log level to <code>ERROR</code> , the loader controller will only log <code>ERROR</code> and <code>FATAL</code> messages.

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 <code>PERF</code> 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: <code>plcclient.sh -l log_level</code> Windows: <code>plcclient -l log_level</code> where <ul style="list-style-type: none"> <code>log_level</code> 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.
	<h4>Note:</h4> <p>You must run this command on the local host running the <code>plc</code> service.</p>

Loader controller

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 (pl c) service.

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: `$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, the specific loader controller configuration file (`plc_*.xml`), and the default behavior:

LSF host data loaders (`plc_coreutil.xml`)

Data loader name	Data type	Data gathering interval	Data loads to	Loader type
Host core utilization (<code>hostcoreutil loader</code>)	core utilization	5 minutes	HOST_CORE_UTILIZATION	polling

LSF job data loaders (`plc_bj obs-sp012.xml`)

Data loader name	Data type	Data gathering interval	Data loads to	Loader type
Bjobs (<code>lsf bj obs loader</code>)	job-related	15 minutes	LSF_BJOBS	polling

LSF data loaders (`plc_lsf.xml`)

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 (lsfbhostloader)	host utilization and state-related	5 minutes	LSF_BHOSTS	polling
LSF events (lsfeventloader)	events with a job ID, performance events, resource events	5 minutes	LSB_EVENTS LSB_EVENTS_EXECHOSTLIST LSF_PERFORMANCE_METRICS	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_HOSTLIST	polling

LSF advanced data loaders (plc_lsf_advanced.xml)

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

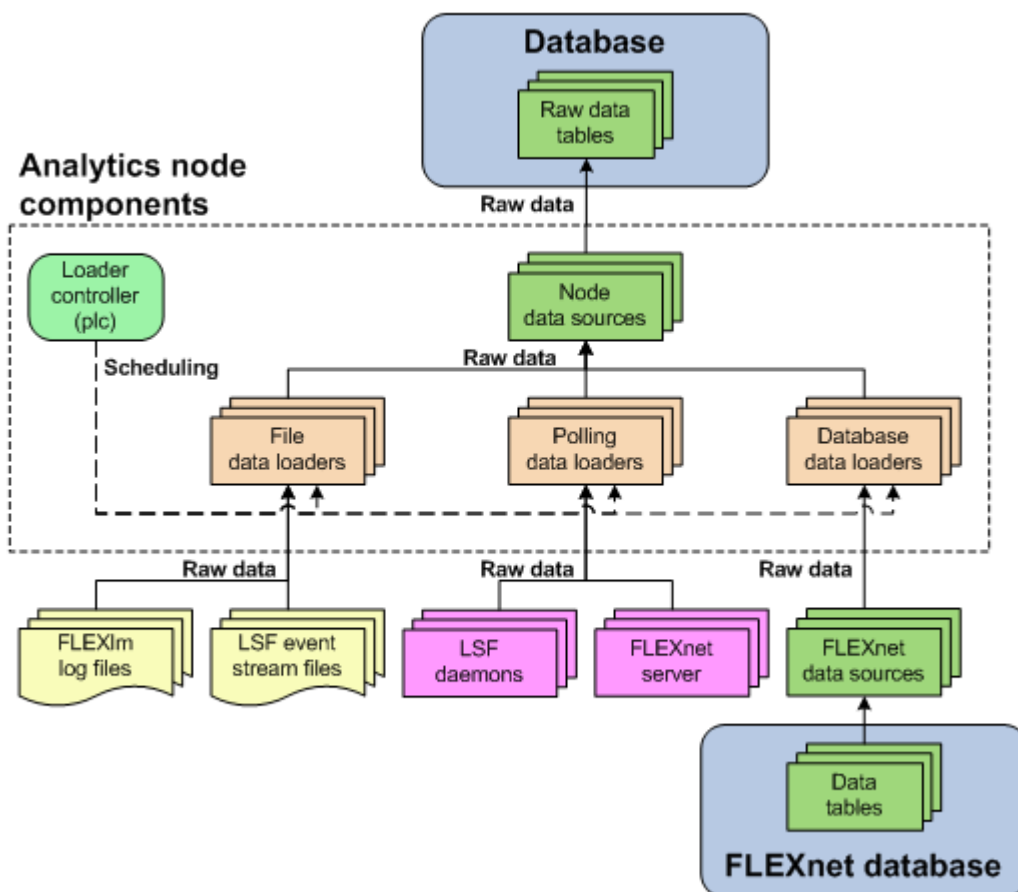
FLEXnet data loaders (plc_license.xml)

Data loader name	Data type	Data gathering interval	Data loads to	Loader type
FLEXnet usage (flexlicenseloader)	license usage	5 minutes	FLEXML_LICENSE_USAGE	polling
FLEXnet events (flexlicenseloader)	license log file event	5 minutes	FLEXML_LICENSE_EVENTS	file
FLEXnet Manager (fmlloader) - only supports FLEXnet Manager 11 or later.	license event	30 minutes	FLEXNET_LICENSE_EVENTS	database

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 (pl c_*. xml) depends on the type of data loader.

These files are located in the loader controller configuration directory:

- UNIX: \$PERF_CONFDIR/pl c
- Windows: %PERF_CONFDIR%\pl c

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 (pl c_*. 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>data_loader_name.xml</i>	<pre><Writer ... EnableRecover="Y"></pre>
Disable data loss protection for the specified data loader.	File location: UNIX: \$PERF_CONFDIR/ data_loader Windows: %PERF_CONFDIR %\data_loader	<pre><Writer ... EnableRecover="N"></pre>

Action	Configuration files	Parameter and syntax
Specify the default log level of your data loader log files.	<p><code>log4j.properties</code></p> <p>File location:</p> <p>UNIX: <code>\$PERF_CONFDIR</code></p> <p>Windows: <code>%PERF_CONFDIR%</code></p>	<pre>log4j.logger.\${data loader}=log_level, \${data loader}</pre> <p>where</p> <ul style="list-style-type: none"> <code>log_level</code> is the default log level of your data loader log files.
Specify the log level of the log files for the specified data loader.		<pre>log4j.logger.data loader.loader_name=log_level</pre> <p>where</p> <ul style="list-style-type: none"> <code>loader_name</code> is the name of the data loader. <code>log_level</code> is the log level of the specified data loader. <p>For example, to set the LSF events data loader (<code>lsfeventsloader</code>) to ERROR, add the following line to <code>log4j.properties</code>:</p> <pre>log4j.logger.data loader.lsfeventsloader=ERROR</pre>
Specify the log level of the log files for the reader or writer area of the specified data loader.		<pre>log4j.logger.data loader.loader_name.area=log_level</pre> <p>where</p> <ul style="list-style-type: none"> <code>loader_name</code> is the name of the data loader. <code>area</code> is either reader or writer. <code>log_level</code> is the log level of the specified data loader. <p>For example, to set the LSF events data loader (<code>lsfeventsloader</code>) writer to DEBUG, add the following line to <code>log4j.properties</code>:</p> <pre>log4j.logger.data loader.lsfeventsloader.writer=ERROR</pre>

The data loaders only log messages of the same or lower level of detail as `log_level`. Therefore, if you change the log level to ERROR, the data loaders will only log ERROR and FATAL messages.

Data loader actions

Action	Command line
View the status and logging levels of the data loaders.	<p>UNIX: <code>plcclient.sh -s</code></p> <p>Windows: <code>plcclient -s</code></p>

Action	Command line
Dynamically change the log level of your data loader log files (temporarily).	UNIX: plcclient.sh -n loader_name -l log_level Windows: plcclient -n loader_name -l log_level 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.
Dynamically change the log level of the log files for the reader or writer area of the specified data loader (temporarily).	UNIX: plcclient.sh -n loader_name -l log_level -a area Windows: plcclient -n loader_name -l log_level -a area where <ul style="list-style-type: none"> • <i>loader_name</i> is the name of your data loader • <i>area</i> is either reader or writer. • <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 (*flexnetcusageloader* and *flexneteventsloader*).
 - **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.
 - **Reader Area:** The reader area of the data loader logs messages of a level specified here and higher. Specify *Inherit* to use the same log level as the entire data loader.
 - **Writer Area:** The writer area of the data loader logs messages of a level specified here and higher. Specify *Inherit* to use the same log level as the entire data loader.
 - **Description:** A description of the data loader.
5. To save any changes and close the window, click OK.

Disable loading of specific data not in use

You can reduce database performance and bandwidth overhead by disabling the loading of specific data that is not in use. You can disable the loading of specific data for the following data loaders:

- Shared resource usage (sharedresourceusage loader): You can disable the *SHARED_RESOURCE_USAGE_HOSTLIST* table.
 - LSF events (lsfevents loader): You can disable the *LSF_PERFORMANCE_METRIC* and *LSB_EVENTS_EXECHOSTLIST* tables, and you can also disable some columns in the *LSB_EVENTS* table.
1. Navigate to the data loader configuration directory.
 - UNIX: \$PERF_CONFDIR/dataloader
 - Windows: %PERF_CONFDIR%\dataloader
 2. Edit the properties file for the data loader that you want to modify.
 - To modify the shared resource usage data loader, edit the *sharedresourceusage.properties* file.
 - To modify the LSF events data loader, edit the *lsfevents.properties* file.
 3. For the LSF events data loader, if you want to disable specific columns in a table, navigate to the *INCLUDED_COLUMNS* parameter and specify the table with a comma-separated list of the columns that you want to include. That is, any columns that you do not specify will be disabled. Use a semicolon to separate multiple tables.

```
INCLUDED_COLUMNS=table_name(column_name1, column_name2, ...)
```

Caution:

The default columns specified for *LSB_EVENTS* are required. If you remove these default columns, you will encounter a data insertion error, or your cubes will have incorrect data.

The required columns for *LSB_EVENTS* are *CLUSTER_NAME*, *EVENT_TYPE*, *EVENT_TIME_GMT*, *EVENT_TIME*, *SUBMIT_TIME*, *JOB_ID*, *JOB_ARRAY_INDEX*, *START_TIME*, *END_TIME*, *NUM_PROCESSORS*, *JOB_NAME*, *USER_NAME*, *USER_GROUP*, *JOB_GROUP*, *FROM_HOST*, *COMMAND*, *QUEUE*, *RES_REQ*, *PROJECT_NAME*, *OPTIONS*, *NUM_EXEC_HOSTS*, *JSTATUS*, *EXIT_STATUS*, *RU_ETIME*, *RU_STIME*, *MAX_RMEM*, *MAX_RSWAP*, *CWD*, *OUT_FILE*, *EXEC_HOSTS*, *APPLICATION_TAG*, *JOB_STATUS_STR*, *NUM_ARRAY_ELEMENTS*, *JOB_INDEX_LIST*, *VERSION_NUMBER*, *JOB_PID*.

For example, if you want to include the *MEM* and *SWAP* columns, but disable all other columns that are not required:

```
INCLUDED_COLUMNS=LSB_EVENTS(CLUSTER_NAME, EVENT_TYPE, EVENT_TIME_GMT,
EVENT_TIME, SUBMIT_TIME, JOB_ID, JOB_ARRAY_INDEX, START_TIME, END_TIME,
NUM_PROCESSORS, JOB_NAME, USER_NAME, USER_GROUP, JOB_GROUP, FROM_HOST,
COMMAND, QUEUE, RES_REQ, PROJECT_NAME, OPTIONS, NUM_EXEC_HOSTS, JSTATUS,
EXIT_STATUS, RU_ETIME, RU_STIME, MAX_RMEM, MAX_RSWAP, CWD, OUT_FILE,
EXEC_HOSTS, APPLICATION_TAG, JOB_STATUS_STR, NUM_ARRAY_ELEMENTS,
JOB_INDEX_LIST, VERSION_NUMBER, JOB_PID, MEM, SWAP)
```

4. Navigate to the *EXCLUDED_TABLES* parameter and specify a semicolon-separated list of tables that you want to disable.

```
EXCLUDED_TABLES=table_name[; table_name]
```

For example,

- For the shared resource usage data loader:

EXCLUDED_TABLES=SHARED_RESOURCE_USAGE_HOSTLIST

- For the LSF events data loader:

EXCLUDED_TABLES=LSB_EVENTS_EXECHOSTLIST; LSF_PERFORMANCE_METRIC

5. Save the properties file and restart the loader controller for your changes to take effect.

Data loaders

Analytics node command-line tools

- [*dbconfig*](#) on page 40
- [*perfadmin*](#) on page 41
- [*plclient*](#) on page 42

dbconfig

Configures the node data source.

Synopsis

dbconfig [**add** *data_source_name* | **edit** *data_source_name*]

dbconfig.sh [**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, 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.

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 (pl c).

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.

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 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.

Analytics node configuration files

- [perf.conf](#) on page 44

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

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

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
- Windows: *ANALYTICS_TOP*\work

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

10

Troubleshooting the node

Actions to troubleshoot the Analytics node

- [Change the default log level of your log files](#) on page 50
- [Disable data collection for individual data loaders](#) on page 50
- [Check the status of the loader controller](#) on page 51
- [Check the status of the data loaders](#) on page 52
- [Check the status of the Analytics node database connection](#) on page 52

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 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 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: `$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
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 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 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 pl c configuration files for your data loaders.
 - For host-related data loaders, edit `pl c_ego.xml` and `pl c_coreutil.xml`.
 - For job-related data loaders (LSF data loaders), edit `pl c_lsf.xml` and `pl c_bj_obs-sp012.xml`.
 - For advanced job-related data loaders (advanced LSF data loaders), edit `pl c_lsf_advanced_data.xml`.
 - For license-related data loaders (FLEXnet data loaders), edit `pl c_license.xml`.

These files are located in the LSF environment directory:

- UNIX: `$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 pl c service.

Check the status of the loader controller

1. If you are logged into a UNIX 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 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: `cd $PERF_TOP/version_number/bin`
 - Windows: `cd %PERF_TOP%\version_number\bin`
4. View the status of the loader controller (pl c) 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: `$PERF_LOGDIR`
- Windows: `%PERF_LOGDIR%`

Check the status of the data loaders

1. If you are logged into a UNIX 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 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 data loader subdirectory of the log directory:

- UNIX: `$PERF_LOGDIR/data_loader`
- Windows: `%PERF_LOGDIR%\data_loader`

Check the status of the Analytics node database connection

1. If you are logged into a UNIX 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 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: **cd \$PERF_TOP/version_number/bin**
 - Windows: **cd %PERF_TOP%\version_number\bin**
4. View the status of the node database connection.
 - UNIX: **dbconfig.sh**
 - Windows: **dbconfig**

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 server

Analytics Console

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

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.
Purger	Displays information about each data purger task.
Addons	Displays information about optional features and add-ons for Analytics.

Analytics Console actions

Action	Command line
Start the Analytics Console.	UNIX: <i>ANALYTICS_TOP/bin/runconsole.sh</i> Windows: Start > Programs > Platform Analytics Server > Platform Analytics Console If you are running this command locally from the Analytics server running UNIX, you need to be running X-Windows. If you are running this command remotely, you need to set your display environment. <hr/> 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.

Data transformers

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

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 `data transformer` subdirectory of your Analytics server log directory:

- UNIX: `ANALYTICS_TOP/log/data transformer`
- Windows: `ANALYTICS_TOP\log\data transformer`

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:

Data transformer name	Transformed database tables
flexEventDaily	WI_FLEXEVENTMARTDAILY
flexEventHourly	WI_FLEXEVENTMART WI_LSFFLEXEVENTMART WI_LSFFLEXEVENTMARTHOLDING
hostGroupConfDaily	WI_HOSTGROUPCONFDAILY
hostGroupConfHourly	WI_HOSTGROUPCONF
hostMetricsDaily	WI_HOSTMETRICSAILY
hostMetricsHourly	WI_HOSTMETRICS
hostPropertiesDaily	WI_HOST_BOOLEANRES WI_HOSTMETRICSAILY WI_HOSTPROPERTIES
hostPropertiesHourly	WI_HOST_BOOLEANRES WI_HOSTMETRICS WI_HOSTPROPERTIES
licConsumpDaily	WI_LICENSECONSUMPTMARTDAILY
licDenialByFeatureDaily	WI_LICENSEDENIALBYFEATUREMARTDAILY
licDenialDaily	WI_LICENSEDENIALMARTDAILY

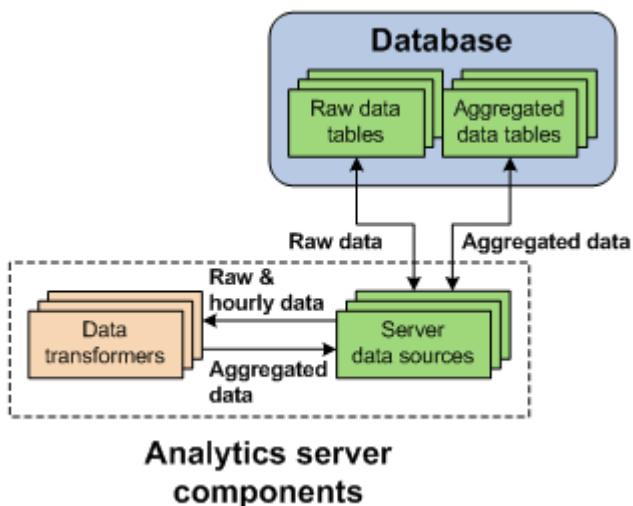
Data transformer name	Transformed database tables
licDenialHourly	WI_LICENSEDENIALMART
licHourly	WI_LICENSECONSUMPTMART WI_LICENSETOTAL WI_LICENSEUTILMART WI_LICUSAGEBYUSERGRPMART
licUsageByUserDaily	WI_LICENSEUSAGEBYUSERMARTDAILY
licUsageByUsrgrpDaily	WI_LICBYUSERGRPMARTDAILY
licUtilByFeatureDaily	WI_LICUTILFEATUREMARTDAILY
licUtilDaily	WI_LICENSEUTILMARTDAILY
IsfDetailPendingReasonHourly	WI_DPR_BYINTERVAL WI_PENDJOBNUM_BYGROUPS WI_PENDTIME_BYGROUPS
IsfDetailPendingReasonJobNumDaily	WI_PENDJOBNUM_BYGROUPSDAILY
IsfDetailPendingReasonTimeDaily	WI_PENDTIME_BYGROUPSDAILY
IsfflexEventDaily	WI_LSFFLEXEVENMART
IsfHostStatusHourly	WI_HOSTSTATUS
IsfJobExecProcHourly	WI_JOBEXECPROCS
IsfJobmartDaily	WI_JOBMA RTDAILY
IsfJobmartHourly	WI_JOBMA RT WI_JOBRESUSAGECOST
IsfJobPendingReasonDaily	WI_JOB SBYPENDREASONDAILY
IsfJobPendingReasonHourly	WI_JOB SBYPENDREASON
IsfJobResUsageByGroupDaily	WI_JOBRESUSAGEBYGROUPDAILY WI_JOB SLOTUSAGEBYGROUPDAILY
IsfJobResUsageByGroupHourly	WI_JOBRESUSAGEBYGROUP WI_JOB SLOTUSAGEBYGROUP
IsfJobResUsageCostDaily	WI_JOBRESUSAGECOSTDAILY
IsfJobStatisticsDaily	WI_NUMBEROFJOBSDAILY
IsfJobStatisticsHourly	WI_NUMBEROFJOBS
IsfJobThroughputDaily	WI_JOBTHROUGHPUTDAILY
IsfJobThroughputHourly	WI_JOBTHROUGHPUT

Data transformer name	Transformed database tables
IsfSharedResourcesDaily	WI_SHAREDRESOURCEUSAGE DAILY
IsfSharedResourcePropertiesHourly	WI_RESOURCEPROPERTIES
IsfSharedResourceUsageHourly	WI_SHAREDRESOURCEUSAGE
userGroupConfHourly	WI_USERGROUPCONF
jobs_v_slotutil	WI_JOBS_VS_SLOTUTIL
cluster_slots	WI_CLUSTER_SLOTS

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.	<p><code>log4j.properties</code></p> <p>File location: <code>ANALYTICS_TOP/conf</code></p> <p><code>log4j.properties</code></p>	<pre>log4j.appender.\${datatransformer} =log_level, \${datatransformer}</pre> <p>where</p> <ul style="list-style-type: none"> <code>log_level</code> is the default log level of your data transformer log files.
Specify the log level of the log file for the specified data transformer.		<pre>log4j.logger.transformer.datatransformer_name=log_level</pre> <p>where</p> <ul style="list-style-type: none"> <code>datatransformer_name</code> is the name of the data transformer. <code>log_level</code> is the log level of your data transformer log file. <p>For example, to set <code>hostMetricsHourly</code> to <code>ERROR</code>, add the following line to <code>log4j.properties</code>:</p> <pre>log4j.logger.transformer.hostMetricsHourly=ERROR</pre>
Specify the log level of the log file for the Extractor or Loader in the ETL flow for the specified data transformer.		<pre>log4j.logger.transformer.datatransformer_name.component=log_level</pre> <p>where</p> <ul style="list-style-type: none"> <code>datatransformer_name</code> is the name of the data transformer. <code>component</code> is the ETL flow component. Use <code>extractor</code> to specify the Extractor and use <code>loader</code> to specify the Loader in the ETL flow. <code>log_level</code> is the log level of your data transformer Extractor or Loader log files. <p>For example, to set the Loader in <code>IsfJobmartHourly</code> to <code>WARN</code>, add the following line to <code>log4j.properties</code>:</p> <pre>log4j.logger.transformer.IsfJobmartHourly.loader=WARN</pre>

Action	Configuration files	Parameter and syntax
Specify the log level of the log file for the specified transformer in the sub-flow of the data transformer.	<p><code>log4j.properties</code></p> <p>File location: <code>ANALYTICS_TOP/conf</code></p> <p><code>log4j.properties</code></p>	<p><code>log4j.logger.transformer.datatransformer_name.subflow.transformer_name=log_level</code></p> <p>where</p> <ul style="list-style-type: none"> <code>datatransformer_name</code> is the name of the data transformer. <code>subflow</code> is the name of the subflow. <code>transformer_name</code> is the name of the specific transformer. <code>log_level</code> is the log level of your transformer log file. <p>If there are spaces in the name of the subflow or transformer, replace the spaces with underscores (<code>_</code>).</p> <p>For example, to set the "HPC Array Indx Splitter" transformer in the HPCJobs subflow of the IsfJobmartHourly data transformer to DEBUG, add the following line to <code>log4j.properties</code>:</p> <p><code>log4j.logger.transformer.IsfJobmartHourly.HPCJobs.HPC_Array_Indx_Splitter=DEBUG</code></p>

The data transformer only logs messages of the same or lower level of detail as *log_level*. Therefore, if you change the log level to ERROR, the data transformer will only log ERROR and FATAL messages.

Data transformer actions

Data transformers are installed as scheduled tasks. Change the schedule of data transformer services as you would for scheduled tasks (see [Scheduled tasks](#) on page 87).

Event notification

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

Configuration to modify event notification behavior

Action	Configuration files	Parameter and syntax
Filter specific event notification emails.	eventfilter.properties File location: <i>ANALYTICS_TOP/conf</i>	Add a new line for each filter. Email notifications that match any one of these lines are filtered out. Regular expressions are supported. For example, if the file contains the following: <pre> Communication timeout Connection reset PLC[0-9]+ has been restarted </pre> The following notifications will be filtered out and you will not receive these emails: Communication timeout PLC10 has been restarted at 12:00:00, Jan. 1, 2010.

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

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.
	<p>Important:</p> <p>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 65.</p>

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: 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
```

Event notification

Data purger

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

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 scheduled tasks on the Analytics server, but are disabled by default.

For non-partitioned data schema, the data purger runs as the following two scheduled tasks:

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.

For partitioned data schema, the data purger runs as the following two scheduled tasks:

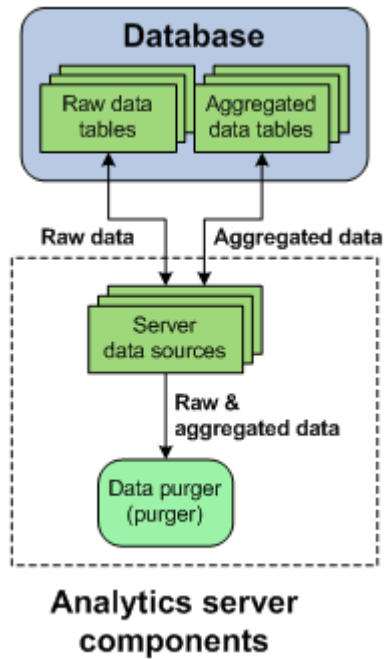
RawTablePartition Purges raw data partitions from both the PERF and Analytics raw data tables. This data purger is disabled by default.

WiTablePartition Purges aggregated data partitions 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.

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. Change the schedule of the data purger service as you would for a scheduled task (see [Scheduled tasks](#) on page 87).

Data purger

15

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.

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 cubes and the datamarts that contributed to the cube build.

Cube	Datamarts that contributed to the cube build
Cluster capacity	Cluster capacity
Daily FLEX license	Daily FLEX license
Hourly FLEX license	Hourly FLEX license
Daily hardware	Daily hardware
Hourly hardware	Hourly hardware
Daily hardware (with boolean resources and core utilization)	Daily hardware (with boolean resources and core utilization) Daily core utilization
Hourly hardware (with boolean resources and core utilization)	Hourly hardware (with boolean resources and core utilization) Hourly core utilization
Daily hardware (with core utilization)	Daily hardware (with core utilization) Daily core utilization
Hourly hardware (with core utilization)	Hourly hardware (with core utilization) Hourly core utilization
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

Cube	Datamarts that contributed to the cube build
Hardware performance prediction	Hardware performance prediction Host properties
Hardware histogram	Hardware histogram
Health check	Health check
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
Jobs vs. slot utilization	Jobs vs. slot utilization
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
License usage by feature histogram	License usage by feature histogram
License usage by server histogram	License usage by server histogram
Daily license usage by user	Daily license usage by user
Hourly license usage by user	Hourly license usage by user

Cube	Datamarts that contributed to the cube build
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 usage statistics	Daily license usage statistics - consumption Daily license usage statistics - denial Daily license usage statistics - utilization
Hourly license usage statistics	Hourly license usage statistics - consumption Hourly license usage statistics - denial Hourly license usage statistics - utilization
Daily license usage statistics by feature	Daily license usage statistics by feature - consumption Daily license usage statistics by feature - denial Daily license usage statistics by feature - utilization
Hourly license usage statistics by feature	Hourly license usage statistics by feature - consumption Hourly license usage statistics by feature - denial Hourly license usage statistics by feature - utilization
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

Logging levels

There are logging levels that determine the detail of messages that the datamarts 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 datamart log files are located in the `datamart` subdirectory of your Analytics server log directory:

- UNIX: `ANALYTICS_TOP/log/datamart`
- Windows: `ANALYTICS_TOP\log\datamart`

Default behavior

Analytics always rebuilds a datamart from the database when a cube is built. You can enable datamart building directly from CSV files, which does not query the database.

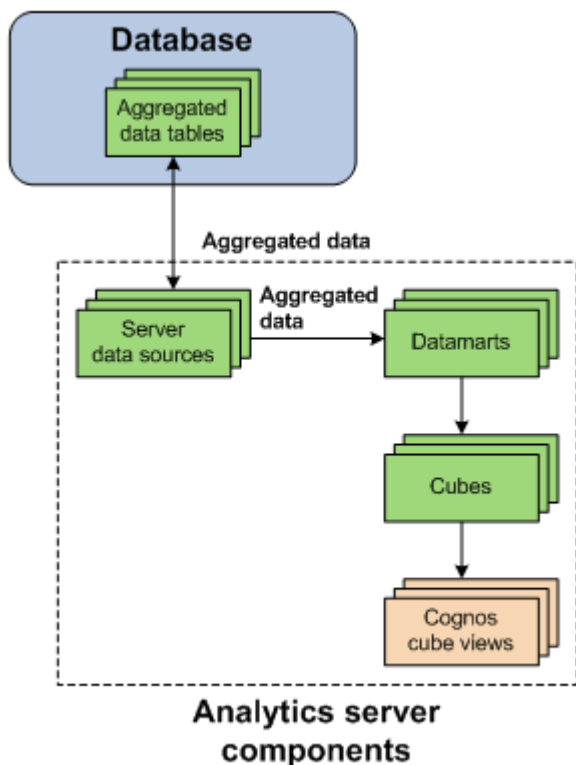
Datamarts and cubes are built incrementally: When building a datamart or cube, only the data of the last day or week for the datamarts will be built and merged with the older data. This improves the Analytics server performance because when incremental datamart building is disabled, the datamarts are built from beginning to end each time.

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.



Configuration to modify datamart behavior

Action	Configuration files	Parameter and syntax
Specify the log level of the log file for the specified datamart.	log4j.properties File location: ANALYTICS_TOP/conf	<pre>log4j.logger.datamart.datamart_name= log_level</pre> <p>where</p> <ul style="list-style-type: none"> • <i>datamart_name</i> is the name of the datamart. • <i>log_level</i> is the log level of the datamart log file. <p>If there are spaces in the name of the datamart, replace the spaces with underscores (_)</p> <p>For example, to set the Hourly Job Information datamart to ERROR, add the following line to log4j.properties:</p> <pre>log4j.logger.datamart.Hourly_Job_Information= ERROR</pre>
Specify the log level of the log file for the specified datamart if you manually build it.		<pre>log4j.logger.datamart.manualBuild. datamart_name=log_level</pre> <p>where</p> <ul style="list-style-type: none"> • <i>datamart_name</i> is the name of the datamart. • <i>log_level</i> is the log level of the datamart log file. <p>If there are spaces in the name of the datamart, replace the spaces with underscores (_)</p> <p>For example, to set the Hourly Job Information datamart to WARN, add the following line to log4j.properties:</p> <pre>log4j.logger.datamart.manualBuild.Hourly_Job Information=WARN</pre>
Specify the log level of the log file for the specified transformer in the datamart.		<pre>log4j.logger.datamart.datamart_name. transformer_name=log_level</pre> <p>where</p> <ul style="list-style-type: none"> • <i>datamart_name</i> is the name of the datamart. • <i>transformer_name</i> is the name of the specific transformer. • <i>log_level</i> is the log level of the transformer log file. <p>If there are spaces in the name of the datamart or transformer, replace the spaces with underscores (_)</p> <p>For example, to set the "SplitQueueTime" transformer in the Hourly Job Information datamart to DEBUG, add the following line to log4j.properties:</p> <pre>log4j.logger.datamart.Hourly_Job_Information. SplitQueueTime=DEBUG</pre>

Action	Configuration files	Parameter and syntax
Specify the log level of the log file for the specified transformer in the datamart if you manually build it.	<p><code>log4j.properties</code></p> <p>File location: <code>ANALYTICS_TOP/conf</code></p>	<p><code>log4j.logger.datamart.manualBuild.datamart_name.transformer_name=log_level</code></p> <p>where</p> <ul style="list-style-type: none"> <code>datamart_name</code> is the name of the datamart. <code>transformer_name</code> is the name of the specific transformer. <code>log_level</code> is the log level of the transformer log file. <p>If there are spaces in the name of the datamart or transformer, replace the spaces with underscores (<code>_</code>)</p> <p>For example, to set the "SplitQueueTime" transformer in the Hourly Job Information datamart to TRACE, add the following line to <code>log4j.properties</code>:</p> <p><code>log4j.logger.datamart.manualBuild.Hourly_Job_Information.SplitQueueTime=TRACE</code></p>
Specify the log level of the log file for the Extractor or Loader in the specified data transformer.		<p><code>log4j.logger.datamart.datamart_name.component=log_level</code></p> <p>where</p> <ul style="list-style-type: none"> <code>datamart_name</code> is the name of the datamart. <code>component</code> is the name of the ETL flow component. Use <code>extractor</code> to specify the Extractor and use <code>loader</code> to specify the Loader in the ETL flow. <code>log_level</code> is the log level of the transformer log file. <p>For example, to set the Extractor in the Hourly Job Information datamart to DEBUG, add the following line to <code>log4j.properties</code>:</p> <p><code>log4j.logger.datamart.Hourly_Job_Information.extractor=DEBUG</code></p>
Enable the incremental datamart building feature (for all datamarts). This is the default behavior.	<p><code>pi.conf</code></p> <p>File location: <code>ANALYTICS_TOP/conf</code></p>	<p><code>INCREMENTAL_DATAMART_BUILD=YES</code></p>
After enabling this feature, you can enable or disable it for an individual datamart by editing that datamart (see Edit or view a datamart on page 78).		
Disable the incremental datamart building feature (for all datamarts).		<p><code>INCREMENTAL_DATAMART_BUILD=NO</code></p>

Action	Configuration files	Parameter and syntax
Disable building the datamart directly from CSV files (that is, enable building datamarts from the database). This is the default behavior	pi . conf File location: <i>ANALYTICS_TOP/conf</i>	DATAMART_BUILD_FROM_FILES=NO
Enable building the datamart directly from CSV files.		DATAMART_BUILD_FROM_FILES=YES

The datamart only logs messages of the same or lower level of detail as *log_level*. Therefore, if you change the log level to ERROR, the datamart will only log ERROR and FATAL messages.

Datamart actions

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 view or edit the datamart build properties, click the ETL tab.
 - To modify the duration of the datamart (that is, the data range of the data in the datamart), modify the Date Selection Range fields.
 - To disable incremental datamart building for this datamart, deselect the Incremental Datamart Build field (or select this field to enable it).

Caution:

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

16

Cubes

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

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
Cluster capacity	Cluster capacity
Daily FLEX license	Daily FLEX license
Hourly FLEX license	Hourly FLEX license
Daily hardware	Daily hardware
Hourly hardware	Hourly hardware
Daily hardware (with boolean resources and core utilization)	Daily hardware (with boolean resources and core utilization) Daily core utilization
Hourly hardware (with boolean resources and core utilization)	Hourly hardware (with boolean resources and core utilization) Hourly core utilization
Daily hardware (with core utilization)	Daily hardware (with core utilization) Daily core utilization
Hourly hardware (with core utilization)	Hourly hardware (with core utilization) Hourly core utilization
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
Hardware histogram	Hardware histogram
Health check	Health check
Daily host status	Host properties Daily host status

Cube	Datamarts that contributed to the cube build
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
Jobs vs. slot utilization	Jobs vs. slot utilization
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
License usage by feature histogram	License usage by feature histogram
License usage by server histogram	License usage by server histogram
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 usage statistics	Daily license usage statistics - consumption Daily license usage statistics - denial Daily license usage statistics - utilization

Cube	Datamarts that contributed to the cube build
Hourly license usage statistics	Hourly license usage statistics - consumption Hourly license usage statistics - denial Hourly license usage statistics - utilization
Daily license usage statistics by feature	Daily license usage statistics by feature - consumption Daily license usage statistics by feature - denial Daily license usage statistics by feature - utilization
Hourly license usage statistics by feature	Hourly license usage statistics by feature - consumption Hourly license usage statistics by feature - denial Hourly license usage statistics by feature - utilization
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

Logging levels

There are logging levels that determine the detail of messages that the cubes 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 cubes log messages of INFO level or higher (that is, all INFO, WARN, ERROR, and FATAL messages).

The cube log files are located in the cube subdirectory of your Analytics server log directory:

- UNIX: `ANALYTICS_TOP/log/cube`
- Windows: `ANALYTICS_TOP\log\cube`

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

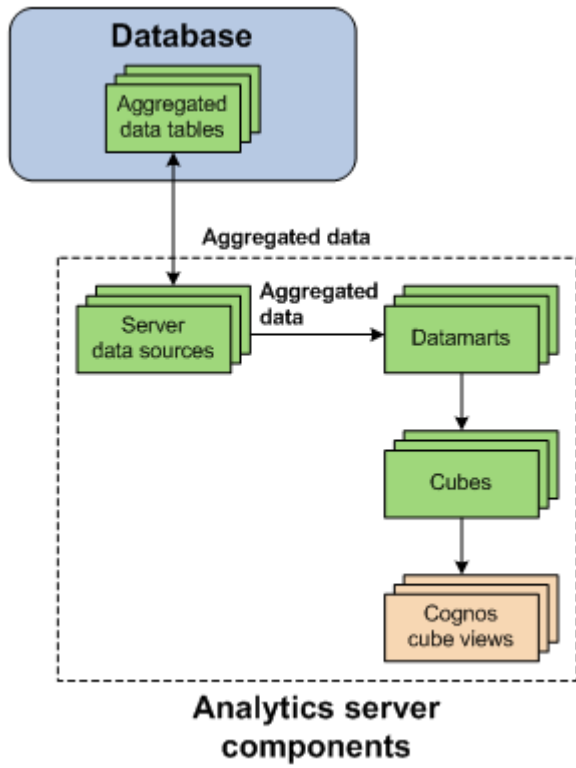
When you open a cube, the first and second dimensions are displayed respectively as rows and columns in the display area.

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.



Configuration to modify cube behavior

Action	Configuration files	Parameter and syntax
Specify the log level of the log file for the specified cube.	<code>log4j.properties</code> File location: <code>ANALYTICS_TOP/conf</code>	<pre>log4j.logger.cube.cube_name=log_level</pre> where <ul style="list-style-type: none"> <code>cube_name</code> is the name of the cube. <code>log_level</code> is the log level of the cube log file. If there are spaces in the name of the cube, replace the spaces with underscores (<code>_</code>) For example, to set the Hourly Workload cube to ERROR, add the following line to <code>log4j.properties</code> : <pre>log4j.logger.cube.Hourly_Workload=ERROR</pre>
Specify the log level of the log file for the specified cube if you manually build it.		<pre>log4j.logger.cube.manualBuild.cube_name=log_level</pre> where <ul style="list-style-type: none"> <code>cube_name</code> is the name of the cube. <code>log_level</code> is the log level of the cube log file. If there are spaces in the name of the cube, replace the spaces with underscores (<code>_</code>) For example, to set the Hourly Workload cube to WARN, add the following line to <code>log4j.properties</code> : <pre>log4j.logger.cube.manualBuild.Hourly_Workload=WARN</pre>

The cube only logs messages of the same or lower level of detail as *log_level*. Therefore, if you change the log level to ERROR, the cube will only log ERROR and FATAL messages.

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

For further details, see [Manage cubes](#) on page 117.

Cubes

Scheduled tasks

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

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 (`BuildCubesHourly`, `BuildCubes`)

The build cubes scheduled tasks build OLAP cubes.

By default, the `BuildCubesHourly` scheduled task builds all hourly cubes at 2:00 a.m. every day, and the `BuildCubes` scheduled task builds all daily cubes (except the Daily Hardware by Host Group and Daily BLD License Usage/Daily Distinct Users and the Weekday Performance cubes) at 3:30 a.m. every day. The cubes are published to Cognos Upfront after they are successfully built.

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

Note:

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.

Improving the `BuildCubesHourly` scheduled task

Since the `BuildCubesHourly` scheduled task runs every day, the hourly cubes only show the data from the last day. To improve this scheduled task so that hourly cubes always show data from the latest hour,

you should separate hourly cubes from the build cubes task into new scheduled tasks, and configure the Analytics server to run these tasks in parallel.

For more details, refer to *Installing Platform Analytics* (specifically, refer to *Improve the hourly build cubes task to show data from the last hour* in the *Optimize the Analytics server* chapter)

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 `builddatamarts.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 57.

Tablespace partition (RawTablePartition and WiTablePartition)

The TablePartition tablespace partition scheduled tasks maintain partitioned tablespaces by creating new tablespaces or deleting tablespaces that are out of date. The RawTablePartition scheduled task maintains raw tablespace partitions and runs at 4:00 a.m. every day by default, while the WiTablePartition scheduled task maintains aggregated tablespace partitions runs at 5:00 a.m. every day.

Tablespace monitor (TSMonitor)

The TSMonitor scheduled task notifies you if any tablespaces have a utilization that exceeds a specified threshold. If you do not increase the tablespace size (or the tablespace does not increase in size automatically), the scheduled task sends a notification every hour. This scheduled task is disabled by default.

This scheduled task has the following prerequisites:

- To monitor the tablespaces, the Analytics user on the Oracle database must have the appropriate privileges in the tablespaces (by selecting the privileges on the `DBA_DATA_FILES` and `DBA_FREE_SPACE` tables).

- Email notification must be enabled to receive the scheduled task notifications. See the Email notification chapter for more details.

Scheduled task actions

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 for detailed information .
View or edit a task from the list of scheduled tasks.	See 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 displays.

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: *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 BuildCubesHourly and BuildCubes scheduled tasks, 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. For the **TSMonitor** scheduled task, specify the details of the tablespaces you want to monitor.

- a) Specify the name of the data source in the **Data Source** field.

The default name of the data source is **ReportDB**.

- b) Specify a comma-separated list of the tablespaces for **TSMonitor** to monitor in the **Tablespaces** field. Leave this field blank if you want **TSMonitor** to monitor all tablespaces in the Oracle database.

For example, to monitor the **TS_DATA_01**, **TS_DATA_02**, and **SYSTEM** tablespaces,

TS_DATA_01, TS_DATA_02, SYSTEM

- c) Specify a threshold for tablespace utilization, as a percentage or a decimal value between 0 and 1, in the **Threshold** field.

If there is at least one monitored tablespace that exceeds this threshold, **TSMonitor** sends a notification every hour until the tablespaces no longer exceed the threshold.

For example, to set a threshold of 90%, specify **0.9** or **90%** in the **Threshold** field.

7. To save your changes and close the window, click **OK**.

Analytics server command-line tools

- [daemons](#) on page 94
- [runBuilder](#) on page 95
- [runconsole](#) on page 97

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.

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 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.

-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 e. sh` is the command for UNIX and `runconsole e` is the command for Windows.

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

Analytics server configuration files

- [pi.conf](#) on page 100

pi.conf

The `pi.conf` file controls the operation of the Analytics server.

About pi.conf

`pi.conf` specifies the configuration of various Analytics server components and features.

Changing pi.conf configuration

After making any changes to `pi.conf`, run the following commands to restart the Analytics server and apply your changes:

- UNIX: From the `ANALYTICS_TOP/bin` directory,

```
daemons.sh stop  
daemons.sh start
```
- Windows: From the `ANALYTICS_TOP\bin` directory,

```
daemons.bat stop  
daemons.bat start
```

Location

The location of `pi.conf` is in `ANALYTICS_TOP/conf`.

Format

Each entry in `pi.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 `#if` as this is reserved syntax for time-based configuration.

PIAM_PORT

Syntax

```
PIAM_PORT=port_number
```

Description

Specifies the Platform Automation Manager listening port number.

Default

9991

CHECK_INTERVAL

Syntax

```
CHECK_INTERVAL=time_in_seconds
```

Description

Specifies the interval, in seconds, that Platform Automation Manager checks the system.

Default

60 seconds

send_notifications

Syntax

send_notifications=true | **false**

Description

Enables event notification.

You would normally configure this parameter using the Analytics Console (in the navigation tree, click Events, then right-click on the list of events and select Action > Notification).

If set to true, 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.

For more information on event notification, refer to [Event notification](#) on page 63.

Default

true

mail.smtp.host

Syntax

mail.smtp.host=*host_name.domain_name*

Description

Specifies the SMTP server that Analytics uses to send event notification emails.

You would normally configure this parameter using the Analytics Console (in the navigation tree, click Events, then right-click on the list of events and select Action > Notification).

Example

```
mail.smtp.host=smtp.example.com
```

Valid values

Any fully-qualified SMTP server name.

Default

Not defined.

from_address

Syntax

from_address=*email_account*

Description

Specifies the sender email address that Analytics uses to send event notification emails.

You would normally configure this parameter using the Analytics Console (in the navigation tree, click Events, then right-click on the list of events and select Action > Notification).

Example

```
from_address=system@example.com
```

Default

Not defined

to_address

Syntax

```
to_address=email_account
```

Description

Specifies the email addresses of the intended recipient of the event notification emails that Analytics will send.

You would normally configure this parameter using the Analytics Console (in the navigation tree, click Events, then right-click on the list of events and select Action > Notification).

Example

```
to_address=admin@example.com
```

Default

Not defined

subject_text

Syntax

```
subject_text=text
```

Description

Specifies the subject of the event notification emails that Analytics will send.

You would normally configure this parameter using the Analytics Console (in the navigation tree, click Events, then right-click on the list of events and select Action > Notification).

Example

```
subject_text=Platform Analytics Error Notification
```

Default

Not defined

message_header

Syntax

message_header=*text*

Description

Specifies the header of the event notification emails that Analytics will send. The rest of the email contains information about the event change and is not specified here.

You would normally configure this parameter using the Analytics Console (in the navigation tree, click Events, then right-click on the list of events and select Action > Notification).

Example

```
message_header=An error has occurred in the Platform Analytics data collection system.
```

Default

Not defined

PIEM_PORT

Syntax

PIEM_PORT=*port_number*

Description

Specifies the Platform Event Manager listening port number.

Default

37600

PIEM_HOST

Syntax

PIEM_HOST=*port_number*

Description

Specifies the Platform Event Manager host.

Default

local host

PIEM_TIMEOUT

Syntax

PIEM_TIMEOUT=*time_in_seconds*

Description

Specifies the timeout, in seconds, for Platform Event Manager to receive events.

Default

36000 seconds (10 hours)

EVENTLOGGER_TIMEOUT

Syntax

EVENTLOGGER_TIMEOUT=*time_in_seconds*

Description

Specifies the timeout, in seconds, for the Platform Event Manager client to send event notifications.

Default

5 seconds

EVENT_LEVEL

Syntax

EVENT_LEVEL=**ALL** | **TRACE** | **DEBUG** | **INFO** | **WARN** | **ERROR** | **FATAL** | **OFF**

Description

Specifies the logging levels of events to send to Platform Event Manager. All events of this specified level or higher are sent. In decreasing level of detail, these are TRACE, DEBUG, INFO, WARN, ERROR, and FATAL.

Use ALL to specify all messages and OFF to specify no messages.

Example

```
EVENT_LEVEL=WARN
```

All WARN, ERROR, and FATAL messages are sent to Platform Event Manager.

Default

INFO

All INFO, WARN, ERROR, and FATAL messages are sent to Platform Event Manager.

DS_NAME

Syntax

DS_NAME=*data_source_name*

Description

Specifies the name of the data source for Platform Event Manager to access.

Default

ReportDB

BatchBuildThreshold

Syntax

BatchBuildThreshold=*integer*

Description

Specifies the number of cubes that can be built in parallel.

Do not specify a value greater than (the number of cores available) - 4, since four cores are assigned to other Analytics tasks. Therefore, do not increase this parameter to 2 unless the Analytics server has six or more cores available.

Valid values

Specify an integer between 1 and 6.

Default

1

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 for cube and datamart building.

The value for LICENSE_FILE can be either of the following:

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

```
LI CENSE_FI LE=/usr/share/l sf/cl uster1/conf/l i cense. dat
```
 - Windows examples:

```
LI CENSE_FI LE= C: \l i censes\l i cense. dat
```

```
LI CENSE_FI LE=\\HostA\l i censes\l i cense. dat
```
- For a permanent license, the name of the license server host and TCP port number used by the l mgrd daemon, in the format *port@host_name*. For example:

```
LI CENSE_FI LE=" 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:

```
LI CENSE_FI LE=" port@hostA: port@hostB: port@hostC"
```

Windows:

```
LI CENSE_FI LE=" port@hostA; port@hostB; port@hostC"
```

Multiple license files should be quoted and must be separated by a pipe character (|).

Windows example:

```
LI CENSE_FI LE="C: \l i censes\l i cense1|C: \l i censes\l i cense2|D: \mydi r\l i cense3"
```

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

INCREMENTAL_DATAMART_BUILD

Syntax

INCREMENTAL_DATAMART_BUILD=YES | Y | NO | N

Description

Specify YES or Y to enable the incremental datamart build feature for all datamarts.

If enabled, when building a datamart or cube, only the data of the last day or week for the datamarts will be built and merged with the older data. This improves Analytics server performance because when incremental datamart building is disabled, the datamarts are built from beginning to end each time.

After enabling this feature, you can enable or disable incremental building for a specific datamart using the Analytics Console by editing the datamart properties (see [Datamarts](#) on page 71).

Default

YES

DATAMART_BUILD_FROM_FILES

Syntax

DATAMART_BUILD_FROM_FILES=YES | Y | NO | N

Description

Specify YES or Y to build datamarts directly from CSV files, which does not query the database.

Specify NO or N to build datamarts from the database.

Default

NO

PURGER_BATCH_SIZE

Syntax

PURGER_BATCH_SIZE=*integer*

Description

Specifies the number of records to purge in each batch.

Valid values

Any positive integer

Default

10000000

SHOW_BUSINESS_INFO

Syntax

SHOW_BUSINESS_INFO=YES | Y | NO | N

Description

Specify YES or Y to enable the Data Collection Nodes page in the Analytics Console to display the following optional columns:

- System Purpose
- Display Description
- Business Area

Default

YES

Analytics server configuration files

P A R T



Managing Analytics

Secure your data and working environment

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

Actions to secure your data and working environment

- [Add user classes and users to Cognos \(Windows\)](#) on page 112
- [Add user classes to Cognos \(Solaris or Linux\)](#) on page 113
- [Add a new user directly from Upfront](#) on page 114
- [Open ports to communicate across firewalls](#) on page 114
- [Modify the database password](#) on page 115
- [Modify the Cognos administrator password](#) on page 115

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.
 - 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.

Note:

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=mypasswd -filename=/usr/cognos/cer5/bin/userclass.txt -filetype=2
```

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

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.

Secure your data and working environment

21

Manage cubes

Manage cubes and Cognos cube views.

Actions to manage cubes

- [Build a cube manually](#) on page 118
- [Save a Cognos cube view](#) on page 118
- [Export a Cognos cube view to a PDF file](#) on page 119
- [Create a bookmark to a Cognos cube view](#) on page 119
- [Modify cube properties](#) on page 119

Build a cube manually

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 `BuildCubes` 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.

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.

Note:

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.

Manage cubes

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.

Troubleshooting the server

Actions to troubleshoot the Analytics server

- [Check the health of the Analytics server](#) on page 122
- [Check the Analytics server log files](#) on page 122
- [Check the status of the Analytics server database connection](#) on page 123

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: **`ANALYTICS_TOP/bin/runconsole.sh`**
 - Windows: Start > Programs > Platform Analytics Server > Platform Analytics 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.
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: *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: *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: *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: *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: *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: *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: *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: *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.

Troubleshooting the server

1. Log into the Analytics server host.
2. Launch the Analytics Console.
 - UNIX: ***ANALYTICS_TOP*/bin/runconsole.sh**
 - Windows: Start > Programs > Platform Analytics Server > Platform Analytics 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.

Index

A

- Add-ons
 - Analytics Console 56
- Analytics Console 55
- ANALYTICS_TOP 12
- architecture 9, 10

B

- BuildCubes 88
- BuildCubesHourly 88
- BuildDatamarts 89

C

- category 83
 - child 83
- Cognos
 - administrator password 115
 - cube views
 - bookmark 119
 - exporting as PDF 119
 - saving 118
 - Upfont
 - users 114
 - user classes 112, 113
 - users 112–114
 - cube views 83
 - bookmark 119
 - exporting as PDF 119
 - saving 118
 - cubes 80
 - Analytics Console 56
 - building manually 118
 - dimensions
 - category 83
 - child category 83
 - metric 83
 - log levels

- manually built 85
- measures 83
- properties 119
- views 83

D

- daemons 94
- data loaders 30
 - bjobs 30
 - core utilization 30
 - data loss protection 33
 - FLEXIm 31
 - FLEXnet 31
 - frequency 33
 - host-related 30
 - job-related 30
 - license 31
 - log files 50
 - log levels
 - default 34
 - reader 34
 - specific data loaders 34
 - temporary 35
 - viewing 34
 - writer 34
 - LSF 30
 - LSF advanced data 31
 - scheduling 26
 - status 52
- data loss protection 33
- data mapping 56
- data purger 68
 - Analytics Console 56
 - log levels 68
- data sources 20
 - Analytics Console 56
 - node
 - status 52

- password 115
- ReportDB 20
 - server
 - status 123
- data transformers 58
 - log levels
 - Extractor 61
 - Loader 61
 - specific data transformers 61
 - transformer in subflow 62
- database 18
 - JDBC connection 20
 - password 115
- datamarts 72
 - Analytics Console 56
 - build
 - from CSV files 78
 - from database 78
 - incremental 77
 - editing 78
 - log levels
 - Extractor 77
 - Loader 77
 - manually built 76
 - specific datamarts 76
 - transformer in datamart 76
 - transformer in manually built datamart 77
 - viewing 78
- dbconfig 40
- dimensions 83
 - category 83
 - child category 83
 - metric 83
- directories 12

E

- events 65
 - Analytics Console 56
 - filtering 65
 - notification 65
 - notification email
 - filtering 64

F

- files
 - license 13
- FLEXIm
 - data loaders 31

- FLEXnet
 - data loaders 31

I

- incremental datamart build 77

J

- JDBC connection 20

L

- LicDataAggDaily 89
- LicDataAggHourly 89
- license 13
 - data loaders 31
 - FLEXIm
 - data loaders 31
 - FLEXnet
 - data loaders 31
 - loader controller 26
 - data collection
 - disabling 50
 - log levels
 - default 27
 - temporary 27
 - status 34
 - log files 50
 - log levels
 - cubes 85
 - data loaders
 - default 34
 - reader 34
 - specific 34
 - temporary 35
 - viewing 34
 - writer 34
 - data purger 68
 - data transformers
 - Extractor 61
 - Loader 61
 - specific 61
 - transformer in subflow 62
 - datamarts
 - Extractor 77
 - Loader 77
 - specific 76

- transformer in datamart 76
- transformer in manually built datamart 77
- default 50
 - loader controller
 - default 27
 - temporary 27
- LSF
 - data loaders
 - advanced data 31
- LSF_TOP 12
- LSFDataAggDaily 89
- LSFDataAggHourly 89

M

- measures 83
- metric 83

N

- node 23
 - Analytics Console 56
 - architecture 10
 - commands
 - dbconfig 40
 - perfadmin 41
 - plcclient 42
 - files
 - perf.conf 44
 - troubleshooting 50

P

- PERF 9
 - directories 12
 - services
 - loader controller 26
- PERF_CONFDIR 12
- PERF_LOGDIR 12
- PERF_TOP 12
- PERF_WORKDIR 12
- perf.conf 44
- perfadmin 41
- pi.conf 100
- plcclient 42
- ports 11
- purger 68
 - Analytics Console 56

- log levels 68
- PurgerAggData 68, 89
- PurgerRawData 68, 89

R

- RawTablePartition 68, 89
- ReportDB 20
- runBuilder 95
- runconsole 97

S

- scheduled tasks 56, 88
 - BuildCubes 88
 - BuildCubesHourly 88
 - BuildDatamarts 89
 - data purger 68
 - data transformers 58
 - LicDataAggDaily 89
 - LicDataAggHourly 89
 - LSFDataAggDaily 89
 - LSFDataAggHourly 89
 - PurgerAggData 68, 89
 - PurgerRawData 68, 89
 - RawTablePartition 68, 89
 - scripts 88
 - TSMonitor 89
 - WiTablePartition 68, 89
- server 53
 - Analytics Console 55
 - architecture 10
 - checking 122
 - commands
 - daemons 94
 - runBuilder 95
 - runconsole 97
 - files
 - pi.conf 100
 - log files 122
 - troubleshooting 122
 - services
 - loader controller 26
 - log files 50
 - status
 - data loaders 52
 - data sources
 - node 52

- server 123
- loader controller 34
- system architecture 10

T

- troubleshooting
 - node 50
 - server 122
- TSMonitor 89

U

- user classes 112, 113
- users 112–114

W

- WiTablePartition 68, 89