Administering Platform Analytics

Platform Analytics
Version 8.0
Release date: May 2011

Platform Computing

Copyright

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

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

We'd like to hear from you

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

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

Document redistribution and translation

Internal redistribution

Trademarks

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

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

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

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

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

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

Intel, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Third-party license agreements

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

Contents

PART

I

About Analytics

CHAPTER

1

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, then assembles it into reports for your analysis. Analytics provides all the tools you need to collect the data, load it into a database, then convert it to reports for your analysis using a ROLAP (relational online analytical processing) tool.

CHAPTER

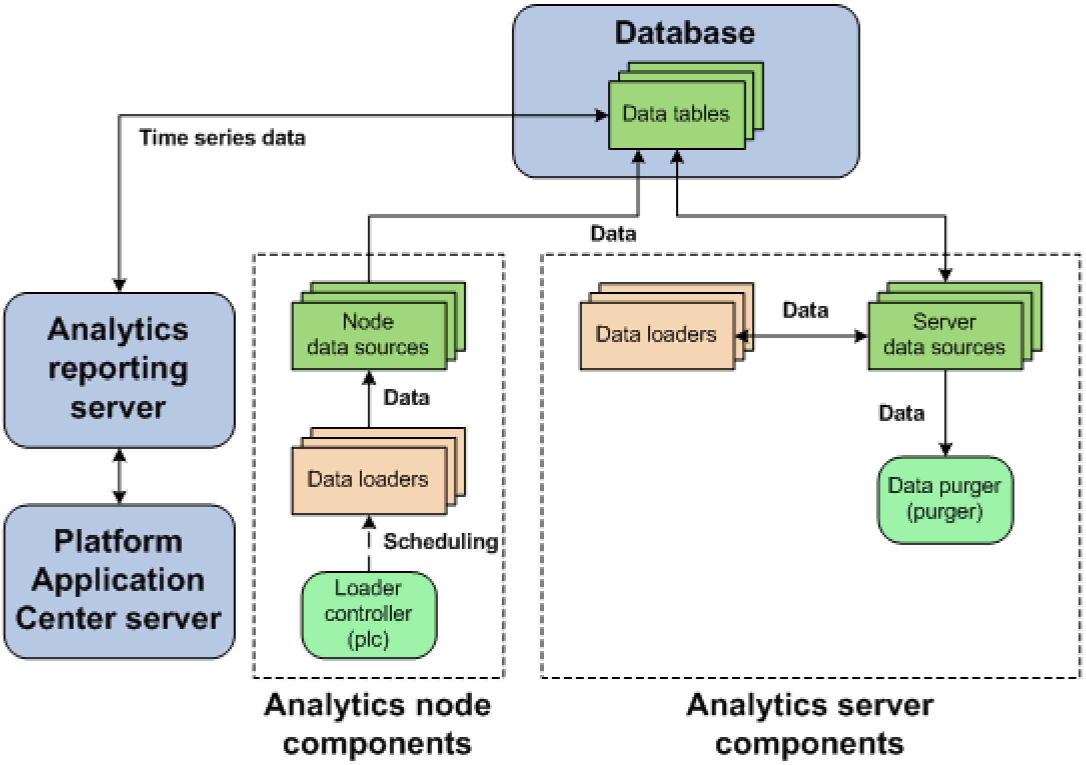
2

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
`$PERF_TOP`	PERF directory	*ANALYTICS_TOP*
`$PERF_CONFDIR`	Configuration files	*ANALYTICS_TOP*/conf
`$PERF_LOGDIR`	Log files	*ANALYTICS_TOP*/log
`$PERF_WORKDIR`	Working directory	*ANALYTICS_TOP*/work

PERF directory environment variables in Windows

Directory name	Directory description	Default file path
`%PERF_TOP%`	PERF directory	*ANALYTICS_TOP*
`%PERF_CONFDIR%`	Configuration files	*ANALYTICS_TOP*\conf
`%PERF_LOGDIR%`	Log files	*ANALYTICS_TOP*\log
`%PERF_WORKDIR%`	Working directory	*ANALYTICS_TOP*\work

CHAPTER

3

Licensing

The Analytics license file includes licenses for data collection.

Contact Platform Computing to obtain a license for Analytics. You may purchase and enable the following components for your Analytics installation to be included in the Analytics license file:

Analytics base	This is a must have license for Platform Analytics. This license allows you to collect data from LSF clusters.
LSF advanced data collection	This 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	This license allows you to collect license usage and event data from your FLEXnet servers.
Vertica database connector	This license allows you to control Analytics database connector.

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`).

PART

II

Database host

Database host

CHAPTER

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 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.

The database and its data schema are partitioned for Analytics data. A partitioned database has tables divided into multiple, smaller tables. This improves database performance for larger clusters.

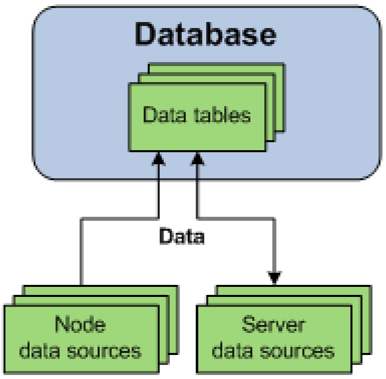
In a large database, purging old job records, transforming data, and other database maintenance tasks can have a significant effect on database performance. Purging old job records and transforming data from smaller tables has less of an impact on the system performance of active tables than on larger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

The database tables are partitioned by quarter. Analytics keeps three years of data in the database. Every month, Analytics has a scheduled task that drops any partition that is older than three years by quarter.

Database interactions

All interactions between Analytics and the database are through the JDBC connection as defined by the data sources.

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



CHAPTER

5

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 data tables in the relational database. The data tables contain processed cluster data that can be extracted and used in reports.

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. 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 data from the data tables through the server data sources, and stores transformed data into the data tables through the server data sources.

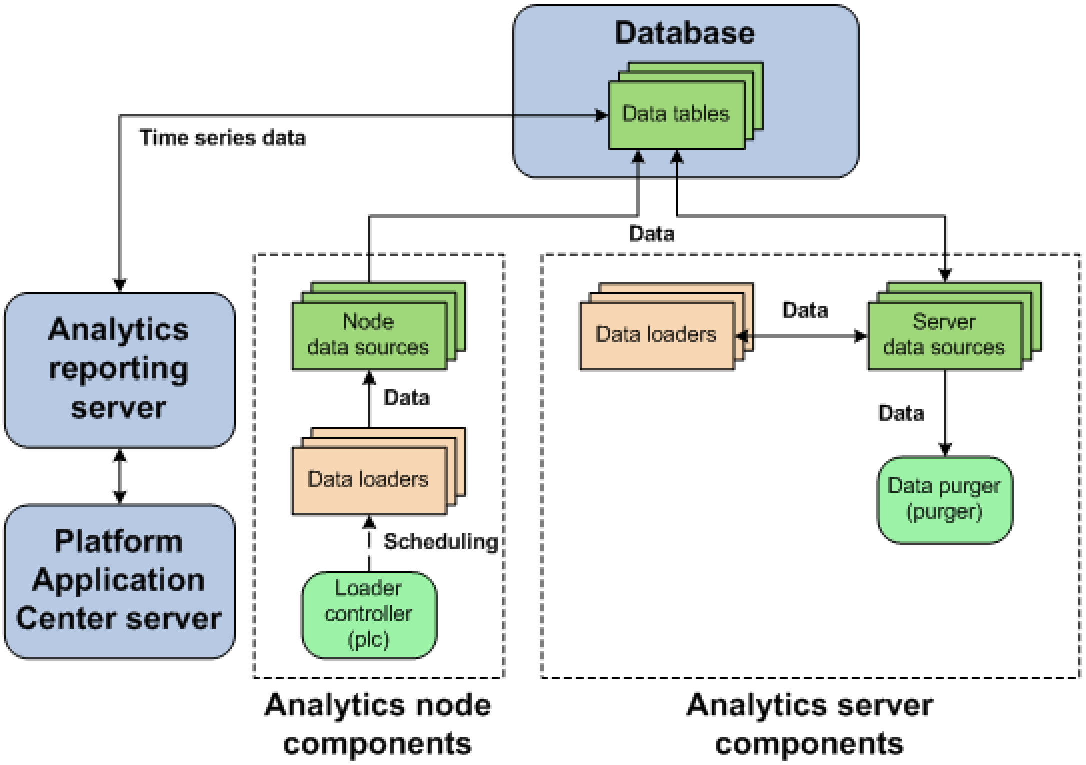
The data purger purges old records from the data tables through the server data sources.

Node data source interactions

The data sources for the Analytics node interact with the 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 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 `perfadmin stop all` and `perfadmin start all`) 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 *data_source_name*** Windows: **dbconfig add *data_source_name*** where • *data_source_name* 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 *data_source_name*** Windows: **dbconfig edit *data_source_name*** where • *data_source_name* is the name the data source that you want to edit.

PART

III

Analytics node

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

Analytics node

CHAPTER

6

Loader controller

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

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%`

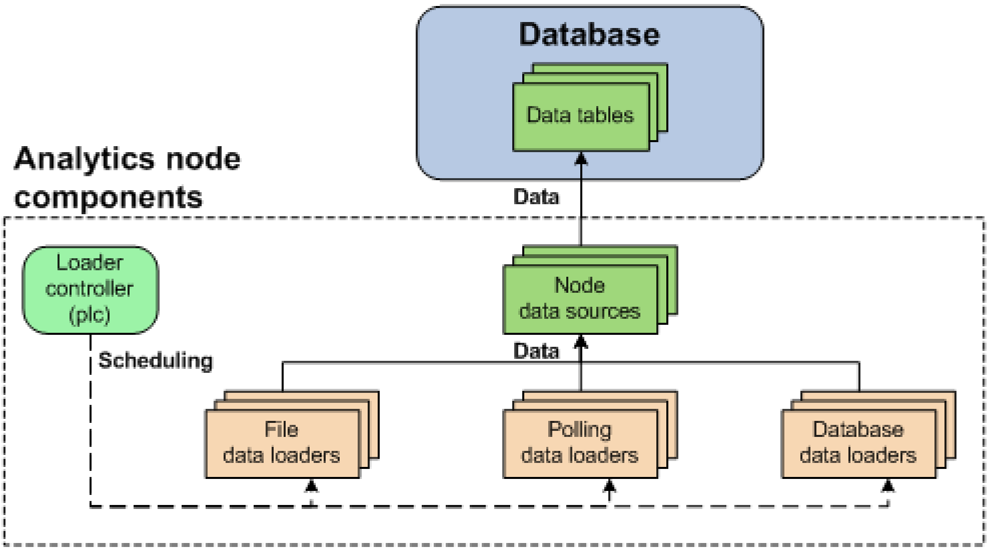
Default behavior

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 data tables through the node data sources while other data loaders obtain it directly from the data tables through the node data sources. The data loaders store this data into data tables through the node data sources. Each data loader contains data that is stored in specific data tables in the database.

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



Configuration to modify loader controller behavior

Action	Configuration files	Parameter and syntax
Specify the default log level of your `plc` log file.	`log4j.properties` File location: UNIX: `$PERF_CONFDIR` Windows: `%PERF_CONFDIR%`	`log4j.logger.com.platform.perf.dataloader=` *`log_level`*`, com.platform.perf.dataloader` where - *log_level* 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 *log_level*. Therefore, if you change the log level to `ERROR`, the loader controller will only log `ERROR` and `FATAL` 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 `plc` and other PERF services.	**perfadmin list**
Stop the `plc` service.	**perfadmin stop plc**
Start the `plc` service.	**perfadmin start plc**

Actions to change the loader controller settings

Action	Command line
Dynamically change the log level of your loader controller log file (temporarily).	UNIX: **plcclient.sh -l** *log_level* Windows: **plcclient -l** *log_level* where - *log_level* 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. **Note:** You must run this command on the local host running the `plc` service.

Loader controller

CHAPTER

7

Data loaders

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

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 dataloader subdirectory of the log directory:

- UNIX: $PERF_LOGDIR/dataloader
- Windows: %PERF_LOGDIR%\dataloader

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 (hostcoreutilloader)	core utilization	5 minutes	HOST_CORE_UTILIZATION	polling

LSF job data loaders (plc_bjobs-sp012.xml)

Data loader name	Data type	Data gathering interval	Data loads to	Loader type
Bjobs (lsfbjobsloader)	job-related	10 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 (lsfbhostsloader)	host utilization and state-related	10 minutes	LSF_BHOSTS	polling
LSF events (lsfeventsloader)	events with a job ID, performance events, resource events	5 minutes	LSB_EVENTS LSB_EVENTS_EXECHOSTLIST LSF_PERFORMANCE_METRIC	file
Resource properties (lsfresproploader)	shared resource properties	1 hour	LSF_RESOURCE_PROPERTIES	polling
SLA (lsfslaloader)	SLA performance	5 minutes	LSF_SLA	polling
Shared resource usage (sharedresusageloader)	shared resource usage	5 minutes	SHARED_RESOURCE_USAGE SHARED_RESOURCE_USAGE_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

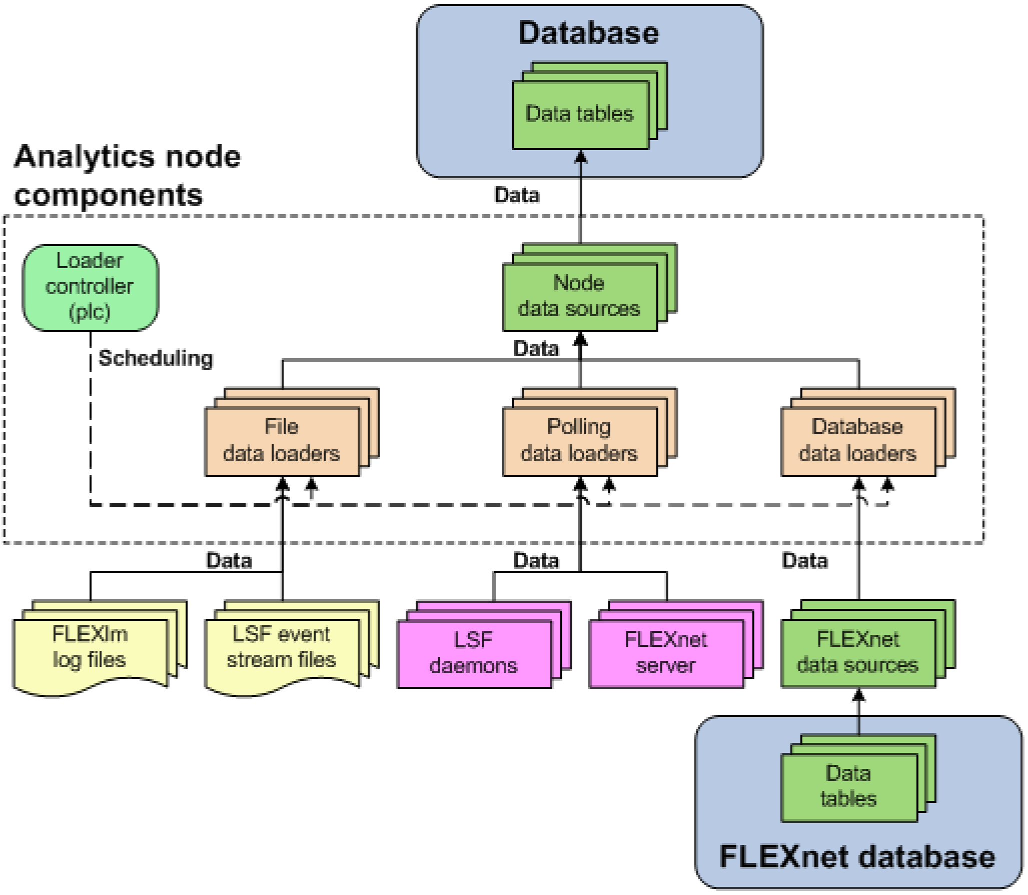
FLEXnet data loaders (plc_license.xml)

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

Data loader interactions

The loader controller service controls the scheduling of the data loaders. The data loaders store LSF and license data into data tables through the node data sources. Each data loader contains data that is stored in specific 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 (`plc_*.xml`) depends on the type of data loader.

These files are located in the loader controller configuration directory:

- UNIX: `$PERF_CONFDIR/plc`
- Windows: `%PERF_CONFDIR%\plc`

Action	Configuration files	Parameter and syntax

Specify the frequency of data gathering for the specified data loader.	Loader controller configuration files for your data loaders (`plc_*.xml`).	`<DataLoader Name="`*`loader_name`*`"`
		`Interval="`*`gather_interval`*`" ... />`
		where
		• *loader_name* is the name of your data loader
		• *gather_interval* is the time interval between data gathering, in seconds

Enable data gathering for the specified data loader.
This is enabled by default.
		`<DataLoader Name="`*`loader_name`*
		`" ... Enable="true" ... />`
		where
		• *loader_name* is the name of your data loader

Disable data gathering for the specified data loader.		`<DataLoader Name="`*`loader_name`*
		`" ... Enable="false" ... />`
		where
		• *loader_name* is the name of your data loader

Enable data loss protection for the specified data loader. This is enabled by default.	Specific data loader configuration file: *`dataloader_name`*`.xml` File location: UNIX: `$PERF_CONFDIR/dataloader` Windows: `%PERF_CONFDIR%\dataloader`	`<Writer ... EnableRecover="Y">`
Disable data loss protection for the specified data loader.		`<Writer ... EnableRecover="N">`

Action	Configuration files	Parameter and syntax
Specify the default log level of your data loader log files.	`log4j.properties` File location: UNIX: `$PERF_CONFDIR` Windows: `%PERF_CONFDIR%`	`log4j.logger.${dataloader}=`*`log_level`*`, ${dataloader}` where • *log_level* is the default log level of your data loader log files.
Specify the log level of the log files for the specified data loader.		`log4j.logger.dataloader.`*`loader_name`*`=`*`log_level`* where • *loader_name* is the name of the data loader. • *log_level* is the log level of the specified data loader. For example, to set the LSF events data loader (`lsfeventsloader`) to `ERROR`, add the following line to `log4j.properties`: **log4j.logger.dataloader.lsfeventsloader=ERROR**
Specify the log level of the log files for the reader or writer area of the specified data loader.		`log4j.logger.dataloader.`*`loader_name`*`.`*`area`*`=`*`log_level`* where • *loader_name* is the name of the data loader. • *area* is either `reader` or `writer`. • *log_level* is the log level of the specified data loader. For example, to set the LSF events data loader (`lsfeventsloader`) writer to `DEBUG`, add the following line to `log4j.properties`: **log4j.logger.dataloader.lsfeventsloader.writer=ERROR**

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.	UNIX: **plcclient.sh -s** Windows: **plcclient -s**

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 • *loader_name* is the name of your data loader • *log_level* 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 • *loader_name* is the name of your data loader • *area* is either `reader` or `writer`. • *log_level* is the log level of your data loader log files. If you restart the loader controller, these settings will revert back to the default level.

View or dynamically edit the data loader settings

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

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

   Note:

   You can only view the data loader properties when the corresponding loader controller is running.
3. Right-click the data loader you want to view or edit and select Properties.
4. Edit the data loader parameters, if needed.

   You can edit the following data loader parameters:

   - Parameters: The specific parameters for the data loader. You can only edit the parameters of FLEXnet data loaders (`flexlicusageloader` and `flexliceventsloader`).
   - Interval (seconds): The data gathering interval of the data loader, in seconds.
   - Log level: The data loader logs messages of a level specified here and higher.
   - 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.

CHAPTER

8

Analytics node command-line tools

dbconfig

Configures the node data source.

Synopsis

UNIX commands:

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

dbconfig.sh -h

Windows commands:

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

dbconfig -h

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 (`plc`).

Options

start *service_name* | all

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

stop *service_name* | all

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

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

UNIX commands:

plcclient.sh [**-s**]

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

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

Windows commands:

plcclient [**-s**]

plcclient [**-l** *log_level*]

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

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.

CHAPTER

9

Analytics node configuration files

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`. If necessary, this default location can be overridden by modifying the PERF_CONFDIR environment variable.

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 `#if` 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 sampling 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 the database 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

PART

IV

Analytics server

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

The server performs the following functions:

- Analytics node management
- Cluster data management

Analytics server

CHAPTER

10

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:

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.

Events
Displays each event logged in Analytics. You can filter the display of these events to find specific events.

Analytics Console actions

Action	Command line
Start the Analytics Console.	UNIX: ***ANALYTICS_TOP*/bin/runconsole.sh** 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. **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.

CHAPTER

11

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 `datatransformer` subdirectory of your Analytics server log directory:

- UNIX: *`ANALYTICS_TOP`*`/log/datatransformer`
- Windows: *`ANALYTICS_TOP`*`\log\datatransformer`

Default behavior

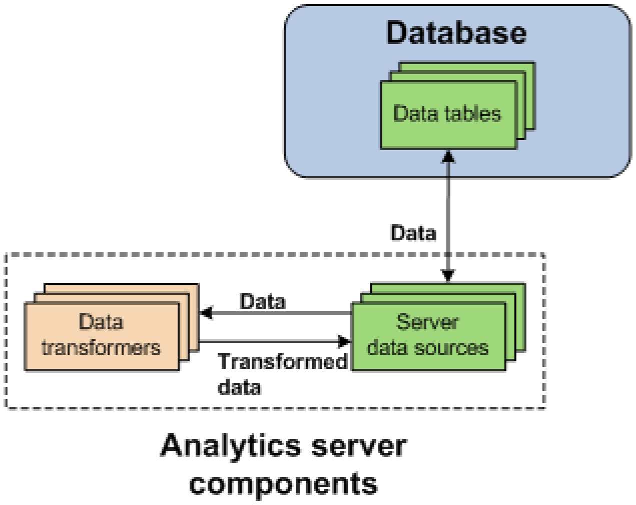
Data transformers convert data at a regular 10 minutes intervals. The following is a list of the data transformers and the database tables in which the data transformers generate the data:

Data transformer name	Transformed database tables
ClusterCapacity	RPT_CLUSTER_CAPACITY_RAW
FlexlmLicUsage	RPT_FLEXLM_LICUSAGE_RAW
Hardware	RPT_HARDWARE_RAW
WorkloadAccounting	RPT_JOBMART_RAW
WorkloadStatistics	RPT_WORKLOAD_STATISTICS_RAW

Data transformer interactions

Data transformers convert raw cluster data from the data tables through the server data sources in the relational database into a format usable for reporting and analysis.

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


Database
Data tables
Data
Data
Data transformers
Transformed data
Server data sources
Analytics server components

Configuration to modify data transformer behavior

Action	Configuration files	Parameter and syntax
Specify the default log level of your data transformer log files.	`log4j.properties` File location: *ANALYTICS_TOP*`/conf` `log4j.properties`	`log4j.appender.${datatransformer}=`*`log_level`*`, ${datatransformer}` where • *log_level* is the default log level of your data transformer log files.
Specify the log level of the log file for the specified data transformer.		`log4j.logger.transformer.`*`datatransformer_name`*`=`*`log_level`* where • *datatransformer_name* is the name of the data transformer. • *log_level* is the log level of your data transformer log file. For example, to set hardware to `ERROR`, add the following line to `log4j.properties`: **log4j.logger.transformer.hardware.loader=ERROR**
Specify the log level of the log file for the Extractor or Loader in the ETL flow for the specified data transformer.		`log4j.logger.transformer.`*`datatransformer_name.component`*`=`*`log_level`* where • *datatransformer_name* is the name of the data transformer. • *component* is the ETL flow component. Use `extractor` to specify the Extractor and use `loader` to specify the Loader in the ETL flow. • *log_level* is the log level of your data transformer Extractor or Loader log files. For example, to set the Loader in WorkloadAccounting to `WARN`, add the following line to `log4j.properties`: **log4j.logger.transformer.WorkloadAccounting.loader=WARN**

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 63).

CHAPTER

12

Event notification

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

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 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. **Important:** If you enable or disable event notification, you need to restart the Platform Task Scheduler to apply this change. See *Restart the Platform Task Scheduler* on page 56.

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 Analytics Task Scheduler 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.

     perfadmin stop all

     perfadmin start all

Configuration to modify event notification behavior

Action	Configuration files	Parameter and syntax

Filter specific event notification emails.	`eventfilter.properties`
	File location:
	ANALYTICS_TOP/conf
		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:
		`Communication timeout`
		`Connection reset`
		`PLC[0-9]+ has been restarted`
		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.

CHAPTER

13

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, which consists of dropping partitions that are older than the calculated data retention date.

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 the following scheduled tasks on the Analytics server:

- `PartitionMaintenanceGroup1`
- `PartitionMaintenanceGroup2`
- `PartitionMaintenanceGroup3`

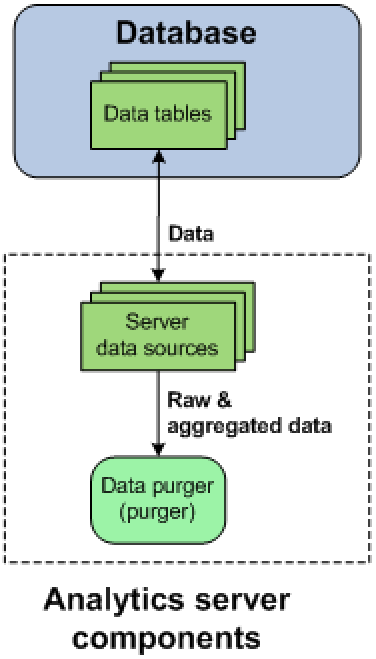
Each scheduled tasks is responsible for purging different tables according to different schedules. This allows the workload to be split among different times.

Each scheduled task calculates the data retention date according to the data purger configuration, examines the tables (and their corresponding partitions) for which it is configured and drops any partitions that are older than the calculated data retention date.

Data purger interactions

The data purger drops database partitions from the 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 scheduled tasks. Change the schedules of the data purger services as you would for scheduled tasks (see *Scheduled tasks* on page 63).

CHAPTER

14

Scheduled tasks

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

About scheduled tasks

After metric data is collected from hosts and stored in the database, the data undergoes several processing tasks for maintenance purposes. 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. These scripts work with data stored in the database for various maintenance tasks such as deleting old or duplicate records, or checking for problems with the collected data.

Predefined scheduled tasks

Analytics includes several predefined scheduled tasks.

Data latency checker (`DataLatencyChecking`)

The data latency checker scheduled task checks the data latency in the data collected from the data loaders and data transformers. If the data latency is longer than the configured value or interval, the data latency checker sends an email notification.

By default, the data latency checker scheduled task runs every hour. If you want to modify the default configuration, edit `ANALYTICS_TOP/conf/health_check_notify.properties` and then restart the Analytics server.

Data purger (`PartitionMaintenanceGroup*`)

The data purger scheduled tasks, which all have "`PartitionMaintenanceGroup`" in their names, control the data purger.

For more information, see *Data purger* on page 59.

Duplicate record remover (`PKViolationClean`)

The duplicate record remover scheduler task checks the most recent data in the database (one to three days old) and deletes any duplicate records in the database (that is, those with a primary key violation). This scheduled task is necessary because the Vertica database does not automatically delete records with a primary key violation.

By default, the duplicate record remover scheduled task runs every 12 hours.

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 *Create, edit, or view a scheduled task* on page 65 for detailed information.

Action	Analytics Console
View or edit a task from the list of scheduled tasks.	See *Create, edit, or view a scheduled task* on page 65 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. To save your changes and close the window, click OK.

CHAPTER

15

Analytics server command-line tools

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 server to control the task scheduler service (`pats`) and the remoting server service (`pars`, if the asynchronous data loading mode is enabled).

Options

start *service_name* | all

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

stop *service_name* | all

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

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.

runconsole

Starts the Analytics console.

Synopsis

runconsole.sh

runconsole

`runconsole.sh` is the command for UNIX and `runconsole` 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.

CHAPTER

16

Analytics server configuration files

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 from the *`ANALYTICS_TOP`*`/bin` directory to restart the Analytics server and apply your changes:

perfadmin stop all

perfadmin start all

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 the 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 55.

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

PIAM_PORT=*port_number*

Description

Specifies the Platform Event Manager host.

Default

`localhost`

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 the 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 the Platform Event Manager to access.

Default

ReportDB

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

PART

V

Support hosts

The support hosts are hosts that run other software to support the operation of Analytics.

While the support hosts do not run Analytics, they are necessary in order for you to take full advantage of the cluster operations data and reports that Analytics assembles and generates.

CHAPTER

17

Analytics reporting server

The Analytics reporting server generates Analytics reports and allows other users to view these reports.

About the Analytics reporting server

The Analytics reporting server runs Tableau Server, which is a ROLAP (Relational Online Analytics Processing) analytic tool for business intelligence that provides browser-based reports. The Analytics reporting server uses Tableau Server to generate the Analytics reports and allows other users to view these reports.

The Analytics reporting server may run on the same host as the Analytics server if that host meets the Tableau Server system requirements.

The Analytics reporting server provides the following default workbooks to allow you to analyze your clusters:

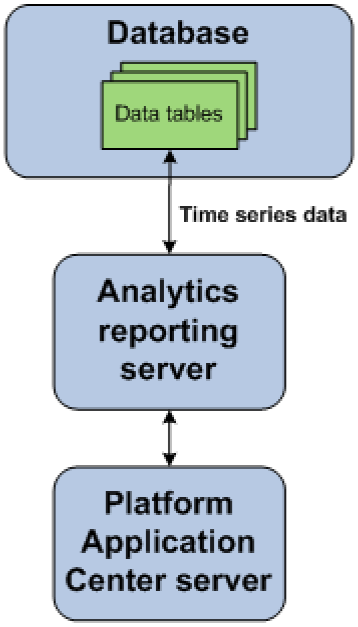
Workbook name	Description
Workload statistics	Reports information about all jobs in any state that are sampled from all active LSF clusters. This allows you to perform a detailed analysis of current LSF workload at any time period.
Workload accounting	Reports job information from LSF job finish events. This allows you to perform a detailed analysis of completed LSF jobs in all clusters.
License usage	Reports on FlexNet Server license usage on any license server or across multiple license servers. This allows you to analyze the usage, consumption, and utilization of licenses by users and hosts.
Hardware	Reports on the hardware utilization at any time period.
Cluster capacity	Reports the usage of all slots in LSF and the workload being run. This allows you to identify IDLE, DOWN, CLOSED, and RUNNING capacity.

If you want to modify or create a new report, use the Platform Analytics designer.

Analytics reporting server interactions

The Analytics reporting server obtains time series data from the database through the Tableau Server data sources. All data obtained by the Analytics reporting server are assembled into reports are then accessible from the Platform Application Center.

The following diagram illustrates the interaction between the Analytics reporting server and other components.


Database
Data tables
Time series data
Analytics reporting server
Platform Application Center server

CHAPTER

18

Platform Application Center host

Platform Application Center allows users and administrators to monitor hosts and to submit and monitor jobs.

About the Platform Application Center host

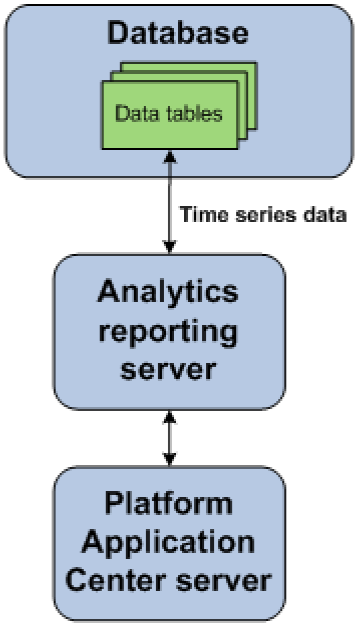
The Platform Application Center host communicates with the Analytics reporting server to provide access to the Analytics reports to monitor the Platform LSF clusters. It also provides browser-based access to all compatible Platform applications.

For more details, refer to the Platform Application Center documentation.

Platform Application Center host interactions

The Analytics reporting server obtains time series data from the database through the Tableau Server data sources. All data that the Analytics reporting server obtains and assembles into reports are then accessible from the Platform Application Center.

The following diagram illustrates the interaction between the support hosts and other components.


Database
Data tables
Time series data
Analytics reporting server
Platform Application Center server

Platform Application Center host

PART

VI

Managing Analytics

CHAPTER

19

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

- *Open ports to communicate across firewalls* on page 92
- *Modify the database password* on page 92

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
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.
Remoting server port (asynchronous data loading mode only)	9093	Internal port for the remoting server. Used for comsmunicating between the remoting server and the remoting node. Configuration is not required. This port is only used if you enabled the asynchronous data loading mode.

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 soruces 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.

CHAPTER

20

Maintaining the database

This section describes the relevant parts in the *Administrator's Guide* for the Vertica Analytic Database that you need to refer to for more details on maintaining the database. All of the following sections are located in the *Operating the Database* chapter of the *Administrator's Guide* for the Vertica Analytic Database.

Actions to maintain the database

- Partition tables in the database.

  You can partition data tables in the Vertica database, which devides one large table into smaller tables. This can optimize query performance by utilizing parallel performance of the disks in which the table partitions reside.

  For more details on recovering the database, refer to *Partitioning Tables* in the *Administrator's Guide* for the Vertica Analytic Database.
- Recover the database.

  You can recover the database to a functional state after at least one node in the system fails.

  For more details on recovering the database, refer to *Recovering the Database* in the *Administrator's Guide* for the Vertica Analytic Database.
- Back up or restore data in the database.

  You can back up or restore data in the database using full backups or incremental backups. You can use backups to recover a previous version

  For more information on backing up or restoring data in the database, refer to Backup and Restore in the *Administrator's Guide* for the Vertica Analytic Database.

CHAPTER

21

Troubleshooting the node

Actions to troubleshoot the Analytics node

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 `plc` configuration files for your data loaders.
   - For host-related data loaders, edit `plc_ego.xml` and `plc_coreutil.xml`.
   - For job-related data loaders (LSF data loaders), edit `plc_lsf.xml` and `plc_bjobs-sp012.xml`.
   - For advanced job-related data loaders (advanced LSF data loaders), edit `plc_lsf_advanced_data.xml`.
   - For license-related data loaders (FLEXnet data loaders), edit `plc_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 `plc` 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 (`plc`) and other PERF services.

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

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

   - UNIX: `$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 `dataloader` subdirectory of the log directory:

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

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**

Check core dump on the Analytics node

Check and enable core dump on the following OS.

Core dump on Linux

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` 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` or `bash`: **. $PERF_TOP/conf/profile.perf**
3. Check if core dump is enabled.

- For `csh` or `tcsh`: **ulimit -c unlimited**
- For `sh` or `bash`: **ulimit -c**

  If it displays 0, then it is disabled.

4. Enable core dump.
   - For `csh` or `tcsh`: **limit coredumpsize unlimited**
   - For `sh` or `bash`: **ulimit coredump**
5. Restart the loader controller and apply your changes.

   perfadmin stop all

   perfadmin start all
6. Collect the stack trace from the node host.
   - Source the environment variables
   - Use gdb to load the core file.

     gdb ${JAVA_HOME}/bin/java core_file

     where *core_file* is the dump core file generated by the Analytics node
   - Print the stack trace: **bt**
7. Collect the output from various installations to check if they are correct.

   For environment variables: **env**

   For `csh` or `tcsh`: **limit**

   For `sh` or `bash`: **ulimit -a**

   Verify rpm packages that you have installed: **rpm -qa|grep glibc**

Core dump on Solaris

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` 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` or `bash`: **. $PERF_TOP/conf/profile.perf**
3. Check if core dump is enabled.
   - For `csh` or `tcsh`: **ulimit -c unlimited**
   - For `sh` or `bash`: **ulimit -c**

   If it displays 0, then it is disabled.
4. Enable core dump.
   - For `csh` or `tcsh`: **limit coredumpsize unlimited**
   - For `sh` or `bash`: **ulimit coredump**
5. Restart the loader controller and apply your changes.

   perfadmin stop all

   perfadmin start all

6. Collect the stack trace from the node host.

   /usr/proc/bin/pstack core_file >pstack.out

   /usr/proc/bin/pmap core_file >pmap.out

   /usr/proc/bin/pldd core_file >pldd.out

   where *core_file* is the dump core file generated by the Analytics node
7. It is recommended that you use `dbx` to collect stack trace.
   - Source the environment variables
   - Use dbx to load the core file.

     dbx ${JAVA_HOME}/bin/java core_file
   - Print the stack trace: **where**
8. Collect the output from various installations to check if they are correct.

   For environment variables: **env**

   For `csh` or `tcsh`: **limit**

   For `sh` or `bash`: **ulimit -a**

   For patches currently installed: **showrev -p**

   For detailed information about the packages installed on a system: **pkginfo -l**

Core dump on AIX and HP-UX

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` 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` or `bash`: **. $PERF_TOP/conf/profile.perf**
3. Check if core dump is enabled.
   - For `csh` or `tcsh`: **ulimit -c unlimited**
   - For `sh` or `bash`: **ulimit -c**

   If it displays 0, then it is disabled.
4. Enable core dump.
   - For `csh` or `tcsh`: **limit coredumpsize unlimited**
   - For `sh` or `bash`: **ulimit coredump**
5. Restart the loader controller and apply your changes.

   perfadmin stop all

   perfadmin start all
6. It is recommended that you use `dbx` to collect stack trace.
   - Source the environment variables
   - Use dbx to load the core file.

     dbx ${JAVA_HOME}/bin/java core_file

where *core_file* is the dump core file generated by the Analytics node

- Print the stack trace: **where**

7. Collect the output from various installations to check if they are correct.

   For environment variables: **env**

   For `csh` or `tcsh`: **limit**

   For `sh` or `bash`: **ulimit -a**

   For release number of the OS: **uname -a**

Debug LSF API

Enable debugging for the LSF API.

1. Set the following environment variables for the current session.
   - For `sh` or `bash`:

     export LSF_DEBUG_CMD="LC_EXEC LC_COMM LC_TRACE"

     export LSF_CMD_LOG_MASK=LOG_DEBUG3

     export LSF_CMD_LOGDIR=`"log_path"`

     export LSB_DEBUG_CMD="LC_EXEC LC_COMM LC_TRACE"

     export LSF_CMD_LOG_MASK=LOG_DEBUG3

     export LSF_CMD_LOGDIR=`"log_path"`

     where `log_path` is the full path where debugging log files are generated.
   - For `tsh` and `tcsh`: Follow the same commands as `sh` or `bash`, but use **setenv** instead of **export**.
2. Restart the loader controller in the same command line session where you set the environment variables.

   perfadmin stop all

   perfadmin start all
3. When data loader start to collect data from LSF, the following log files are generated under the specified directory.
   - `lscmd log` *`host_name`*
   - `bcmd log` *`host_name`*

   Where *host_name* is the name of the Analytics node host.

Analytics node did not respond

If INFO level messages are not updated for more than one hour in the `ANALYTICS_TOP/log/plc.log.`*`host_name`* file, the Analytics node may not respond. Check for the following reasons to resolve this issue.

1. Check if the specified maximum heap size is less than the minimum memory required for the data volume. Check for the following in the log file.

   Memory info before gc: *memory in bytes*

   Memory info after gc: *memory in bytes*

   If the specified heap size is less than the minimum memory requirement, then increase the heap size by changing the java settings in the `ANALYTICS_TOP/conf/wsm/wsm_plc.conf` file.

For example: **JAVA_OPTS=-Xms64m -Xmx2048m**

Note:

For Windows 32bit, the maximum heap size that you can set is 1600M. For Linux / Unix 32bit, you can set it to 4096M. For 64bit machine, you can set it to any value.

2. Check if there is enough disk space for the Analytics node host. If that is the problem, then contact your Administrator to resolve the disk space issue. You need to restart the loader controller once you increase the disk space.

CHAPTER

22

Troubleshooting the server

Actions to troubleshoot the Analytics server

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 Scheduled Tasks in the navigation tree and verify that the scheduled tasks are running correctly according to schedule.

   You can also check the data purger scheduled tasks (`PartitionMaintenanceGroup*`) and compare the data purger settings with your cluster data retention policies.
5. Click Events in the navigation tree and verify that there are no `ERROR` or `FATAL` events.
6. Verify the email notification settings.

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

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 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`
2. 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`
3. 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.

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.

PART

VII

Customizing Analytics

Analytics customizations allow you to maintain and upgrade your Analytics installation to improve performance and fix issues. Contact Platform Support to find more information on specific customizations to meet your needs or to fix specific issues.

Analytics customizations provided by Platform follow specific conventions. If you create your own customizations, your customizations must follow these same conventions to ensure that your customizations are compatible and are saved if you upgrade your Analytics installation.

Naming conventions

The name of the customization is the same as the package name and identifies the specific customization, allowing Platform Support to easily locate the source code for your specific customization.

The customization name is the module or activity name followed by an underscore (_) and a serial number. For Platform customizations, this number is often the support ticket number for the Analytics enhancement or bug fix.

Subdirectories containing files belonging to the customization must have names followed by an underscore and the serial number. Similarly, files belonging to the customization that are located in common directories must also have names followed by an underscore and the serial number.

Node customizations

The following describes conventions and examples of customizations to the Analytics node:

- *Supported files* on page 110
- *Customize an existing data loader* on page 110
- *Add a new custom data loader* on page 111

Supported files

Customizations to the following built-in configuration files (all in the `conf` directory) will remain in the upgraded or patched Analytics node:

- `datasource.xml`
- `log4j.properties`
- `plc.xml`
- `perf.conf`
- All `*.properties` files in the `dataloader` subdirectory.
- All `*.xml` files in the `plc` subdirectory.
- `wsm_plc.conf` files in the `wsm` subdirectory.

Customizations to other Analytics files might not remain in an upgrade or patched Analytics node. Therefore, in order to meet Analytics node conventions, customizations to the Analytics node cannot overwrite any Analytics files not in this supported list.

Customize an existing data loader

If you customize an existing data loader, do not directly overwrite the built-in binaries. Instead, you can edit the source code, make file, or `build.xml` file to build binaries with different names by following the naming conventions.

The following describes an example to customize the `lsfpendingreasonloader` to obtain more information for detailed pending reasons:

1. Edit the necessary source code to change or add the necessary required information.

   For example, edit the `pendreason.c` file.
2. Edit the make file to build the final `.so` file with a different name (such as appending the serial number).

   For example, edit the make file to build the final file named `libpendreason_148781.so`.
3. Change the package name to a different name (such as appending the serial number).

   For example, for all files in the `com.platform.perf.dataloader.lsf.advanced.pendreason` package, change the package name to `com.platform.perf.dataloader.lsf.advanced.pendreason_148781`.
4. Change the Java code to load the new shared library.

   For example, in the `com.platform.perf.dataloader.lsf.advanced.pendreason_148781.ReadPendReasonJNI.java` file, change the `System. loadLibrary` line to the following:

   ```
   System.loadLibrary("pendreason_148781");
   ```
5. Edit the `build.xml` file to build the final `.jar` file with a different name.

For example, edit the `build.xml` file to build the `pendreason_148781.jar` file.

6. Copy the existing data loader configuration to a file that follows the customization file naming convention.

   For example, copy the existing data loader configuration to `pendingreason_148781.xml`.
7. Edit the new data loader configuration file with the desired attributes.
   a) Change the `Class` attribute of the `Reader` element to the new class that you specified as the package name.

      For example, change the Class attribute from `com.platform.perf.dataloader.lsf.advanced.pendreason` to `com.platform.perf.dataloader.lsf.advanced.pendreason_148781`.
   b) To add more columns that you want the data loader to collect, edit the `SQL` section.
8. Edit the loader controller configuration file to point to the new data loader configuration file.

For example, the relevant directories and files are as follows:

`ANALYTICS_TOP`

- `conf`
- `dataloader/pendingreason_148781.xml`

  The data loader configuration file.
- `plc/plc_lsf_advanced.xml`

  The loader controller configuration file related to the pending reason data loader. This file may be modified for the new data loader.
- `lsf/7.0`

  Library files collecting Platform LSF 7.0 data.

  Similarly, the `ego` directory contains library files collecting EGO-related data, and the `license` directory contains library files collecting license-related data.
- `dataloader/pendingreason_148781.xml`

  The data loader configuration file.
- *`platform`*`/lib/libpendreason_148781.so`

  The shared library file is here.

Add a new custom data loader

Add a new data loader to collect custom data from the cluster.

1. Add the loader controller configuration file for the new data loader to the `ANALYTICS_TOP/conf/plc` directory.

   Create a new loader controller configuration file by copying the `plc.xml` file and editing the copied file for your new data loader. It is recommended that you create at least one standalone loader controller configuration file for your custom data loaders.
2. Add the new data loader configuration file to the `ANALYTICS_TOP/conf/dataloader` directory.
3. Add the library files to the corresponding `lib` directories.

For example, to create the License Scheduler workload data loader with serial number 148782, add the following files to the following relevant directories:

`ANALYTICS_TOP`

- `conf`
- `dataloader/ls_workload_148782.xml`

  The data loader configuration file.
- `dataloader/ls_workload_148782.properties`

  The data loader property file.
- `plc/plc_ls_workload_148782.xml`

  A standalone loader controller configuration file for the new data loader.
- `license/7.0`

  Library files collecting Platform LSF License Scheduler 7.0 data.

  Similarly, the `ego` directory contains library files collecting EGO-related data, and the `lsf` directory contains library files collecting LSF-related data.
- `lib/ls_workload_148782.jar`
- *`platform`*`/lib/liblsworkload_148782.so`

  The shared library file is here.

Server customizations

The following describes conventions and examples of customizations to the Analytics server:

- *Supported files* on page 113
- *Customize an existing Tableau Server workbook* on page 113

Supported files

Customizations to the following built-in configuration files (all in the `conf` directory) will remain in the upgraded or patched Analytics server:

- `datasource.xml`
- `log4j.properties`
- `Config.xml`
- `ItemLists.xml`
- `pi.conf`
- All `*.xml` files in the `purger` subdirectory.
- `Package.xml` files in the `packages/workload` subdirectory.

Customizations to other Analytics files might not remain in an upgrade or patched Analytics server. Therefore, in order to meet Analytics server conventions, customizations to the Analytics server cannot overwrite any Analytics files not in this supported list.

Customize an existing Tableau Server workbook

Customizing an existing Tableau Server workbook is not recommended, because the customization is not guaranteed to remain in the upgraded or patched workbook. Instead, copy the existing workbook to a new one following the naming convention. Use the Analytics Desktop to customize the new workbook and publish.

Database schema customizations

When customizing the database schema, you should only perform the following actions:

- Create a new object.
- Add a new column to a built-in table.

Do not perform the following actioons to customize the database schema:

- Drop a built-in object.
- Rename a built-in object.
- Drop a column from a built-in table.
- Rename a column in a built-in table.
- Replace a built-in view, procedure, package, or trigger.

Built-in objects include tables, views, procedures, packages, indexes, triggers, and sequences.

Customization management

The following describes the conventions when assembling, installing, or viewing the customization packages (or "patches").

Assemble the customization package

Binary or configuration files in the customization package should keep the same hierarchical structure as it does in the runtime environment. Perform the following to make your customization package compatible with the Analytics patch installer: and add the following text files to this subdirectory:

1. Create a subdirectory named `patch_install` in the top-level directory of your package.
2. Add patch configuration files to the `patch_install` subdirectory.

   a) Create and add the `patchinfo.txt` file.

   Specify a semicolon-separated list containing detailed patch information in the following format:

   build_number;*build_date*;*version*;*dependency*;*manual_config*

   where

   - *build_number* is the build request number. This is a unique number that distinguishes the patch from other patches. For customizations, specify any unique build number or use a serial number according to the customization naming conventions. For example, `12345`.
   - *build_date* is the build date in UTC/GMT time in the following numerical format: `YYYYMMDDhhmmss`. For example, `20111015104104`.
   - *version* is the version of your Analytics installation. For example, `8.0`.
   - *dependency* is the build number of a fix or solution that this patch depends on. If there is more than one fix or solution dependency, separate multiple build numbers with a comma. If there are no depencies, use `null`. For example, `1234,2345`.
   - *manual_config* specifies if the patch has manual configuration steps before starting the Analytics services. If set to `Y`, the patch installer does not restart Analytics services after deploying the patch; otherwise, the patch installer will restart the Analytics services after deplying the patch. The default value is `N`.

   For example,

   ```
   12345;20111015104104;8.0;1234,2345;Y
   ```

   b) Create and add the `fixlist.db` file.

   Specify a list of bugs fixed in the patch, with each fixed bug on one line in the file. Each line contains the bug tracking number and an optional brief description, ending with a semicolon, as follows:

   bug_number[:*description*];

   For example,

   ```
   148781:Added more columns to pendreasonloader;
   ```

   c) Create and add the `filelist.db` file.

   Specify a list fo files in your customization. Use a slash (`/`) in the file paths for both Windows and UNIX.

For example,

```
conf/dataloader/pendingreason_148781.xml
conf/plc/plc_lsf_advanced.xml
lsf/7.0/lib/pendreason_148781
lsf/7.0/linux_64-x86/lib/libpendreason_148781.so
```

Install the customization package

1. Navigate to the `ANALYTICS_TOP/patch_tools` directory.
2. Run the Analytics server patch installer.
   - UNIX: **patch_install.sh**
   - Windows: **patch_install.bat**

Note:

- The patch installer will prompt you to specify the patch directory, which is the absolute file path to the extracted directory of your patch.
- The patch installer will restart the services on the Analytics server.

View details on the customization packages

The following commands allow you to view information on the customizations that are applied to the Analytics installation.

- List information on all patches applied to the current Analytics installation directory.
  - UNIX: **pversion.sh -a all**
  - Windows: **pversion.bat -a all**

  The latest patch is shown first.
- List information on the last patch that the current file is from.
  - UNIX: **pversion.sh -f** *file_name*
  - Windows: **pversion.bat -f** *file_name*
- List detailed information on the specified build.
  - UNIX: **pversion.sh -b** *build_name*
  - Windows: **pversion.bat -b** *build_name*

Index