Platform Analytics
Version 9.1
for LSF

Administering

SC14-7572-00

Note
Before using this information and the product it supports, read the information in "Notices" on page 91.

First edition

This edition applies to version 9, release 1, modification 0 of Platform Analytics (product number 5725-G84) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Figures

Tables

Chapter 1. About Platform Analytics

IBM® Platform Analytics provides several interactive dashboards that are ready to use "out of the box", making it quick and easy to analyze key data. Existing or new data sources can be rapidly combined with Analytics data to provide data views tailored specifically to an organization's unique requirements without the need to build intermediate data views.

Introduction to Platform Analytics

Platform Analytics is an advanced analysis and visualization tool for analyzing massive amounts of IBM Platform LSF workload data. It enables managers, planners and administrators to easily correlate job, resource and license data from one or multiple clusters for data-driven decision-making. With better insight into the high-performance computing (HPC) data center environment, organizations can identify and quickly remove bottlenecks, spot emerging trends, and plan capacity more effectively.

Unlike traditional business intelligence solutions that require significant time and multiple steps to translate raw data into usable information, Platform Analytics incorporates innovative visualization tools that are built on top of a powerful analytics engine for quick and easy results. You can use the pre-configured dashboards or construct your own, quickly answer questions about your HPC infrastructure and applications, and use that information to optimize HPC resource utilization.

Platform Analytics is a workload intelligence solution for LSF® clusters, FLEXnet license, and FLEXnet Manager license data. Platform Analytics collects LSF and license data, then assembles it into reports for your analysis. Platform 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 relational online analytical processing (ROLAP) tool.

Architecture overview

The Platform Analytics architecture is based on the Platform Enterprise Reporting Framework (PERF) architecture. Platform Analytics adopts and extends the PERF technology to cover all data collection requirements and to improve data collection reliability. For the Analytics database, Platform Analytics supports Analytics, a state-of-the art MPP columnar database that runs on standard hardware and uses a fraction of the resources of traditional database management systems. The Platform Analytics reporting server that has Tableau Server is used as the ROLAP tool to generate reports and to allow other users to view these reports using a web browser.

Major components of Platform Analytics

Figure 1 on page 2 shows the major components of Platform Analytics.

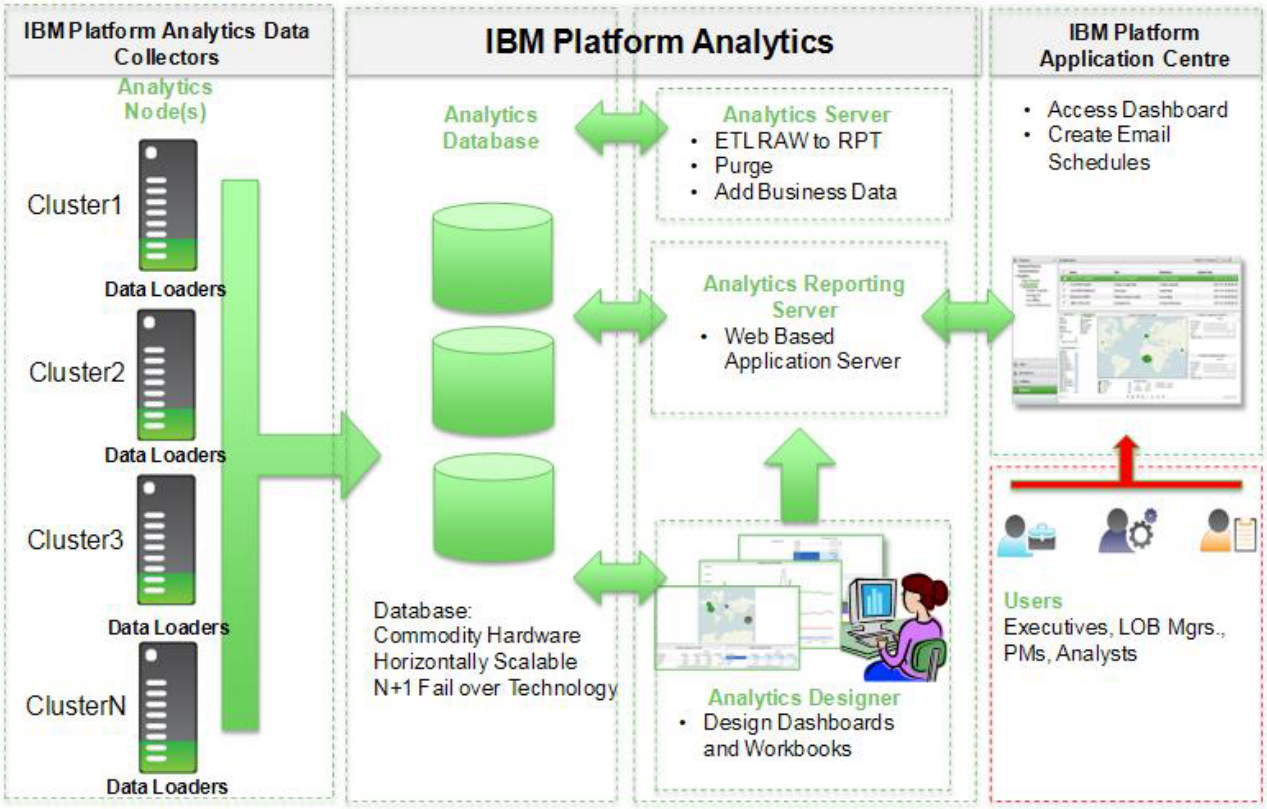

Figure 1. Platform Analytics components

The major Platform Analytics components are:

Platform Analytics Data Collectors for LSF
A data loader is installed on each cluster. The data loader helps to load data directly into the Analytics database. Each data loader collects LSF data, FlexLM License Data (from any number of Flex LM License Servers), and FNM License Data (from a FNM License Server).

Analytics database
Platform Analytics is designed to support the Vertica database, to provide improvements in query and data loading performance over traditional RDBMS technologies. Data is neatly organized into tables for reporting and analysis.

Analytics server
The Analytics server communicates between the data loaders and the Analytics database. It manages the data which the Analytics nodes collect. The Analytics server receives event notification from nodes and other components, and then sends out an email according to the configured rule.

Analytics node
The Analytics node runs data loaders that reliably load data from clusters into the Analytics database

Analytics reporting server
The Analytics reporting server is a web-based reporting tool consisting of workbooks. It collects data from the Analytics database and allows the publishing of dashboards or individual worksheets from Platform Analytics Designer.

Platform Analytics Designer

Platform Analytics Designer provides the flexibility to easily construct complex queries and dashboards specific to each customer's own reporting and analysis requirements. This designer is mainly used for customizing existing Analytics workbooks and for creating new custom workbooks based on Analytics database data.

Platform Application Center

IBM Platform Application Center is used to view the Platform Analytics reports. It allows you to look at the overall statistics of the entire cluster. Platform Application Center helps to analyze the history of hosts, resources, and workload in the cluster to get an overall picture of cluster's performance.

System architecture

Figure 2 shows an overview of the Platform Analytics system architecture and data flow.

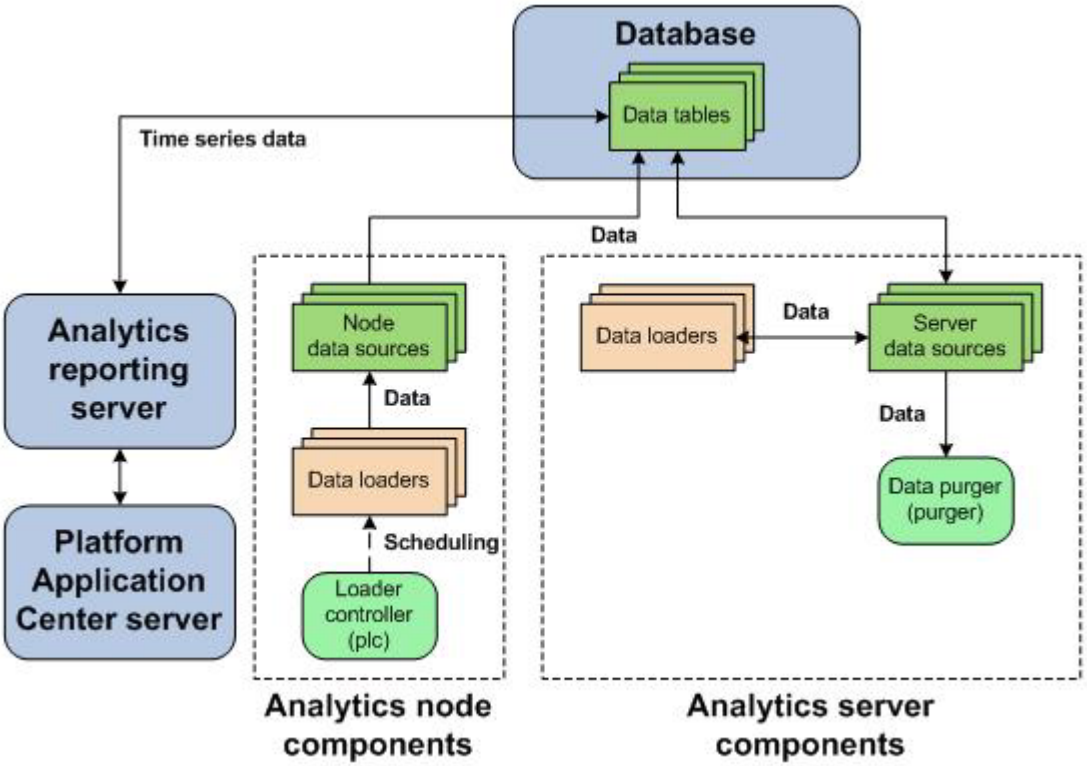

Figure 2. Platform Analytics system architecture and data flow

System ports

For a list of ports that the Platform Analytics hosts use, see *Platform Analytics Installation*, specifically, the "System ports" section in the "Platform 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 Platform Analytics installation directory. In UNIX, you need to source the PERF environment to use these environment variables.

UNIX environment variables for PERF directories

Table 1 lists the UNIX environment variables for PERF directories in the Analytics node.

Table 1. UNIX environment variables for PERF directories

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 2. Managing the database host

The database host includes the database and data sources.

Database

The relational database contains the cluster operations data for reporting and analysis. Platform Analytics components input and output data from the tables within the database.

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

Database interactions

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

Figure 3 illustrates the interaction between the database and other components.

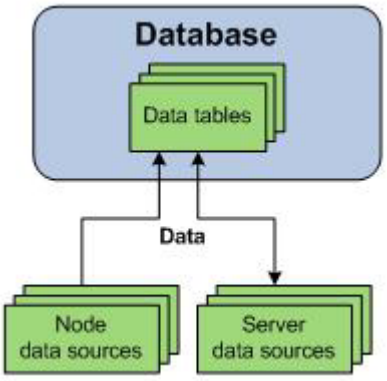

Figure 3. Database and component interaction

Data sources

Data sources define all JDBC connections between the hosts and 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 Platform 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 Platform Analytics accordingly. The default Analytics data source for the server and the node is `ReportDB`.

Platform 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 Platform 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 store 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.

Figure 4 on page 7 illustrates the interaction between data sources and other components.

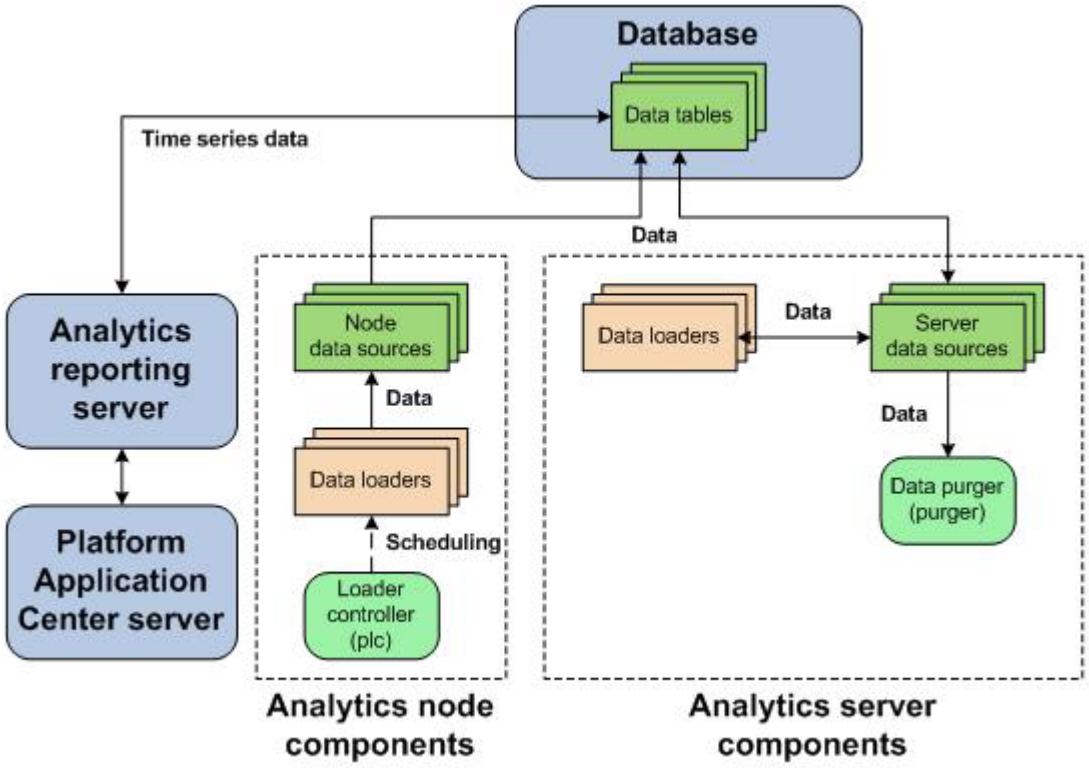

Figure 4. Interactions between data sources and other components

Data source actions

You can perform a variety of actions on the Analytics server data sources and Analytics node data sources.

Actions on the Analytics server data sources

Table 2 lists the actions you can take on the Analytics server data sources.

Table 2. Actions on the Analytics server data sources

Action	Platform 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

Table 3 on page 8 lists the actions you can take on the Analytics node data sources.

If the Analytics node is running on a UNIX host, you must source the Analytics environment before running the **dbconfig.sh** command.

- For **csh** or **tcsh**:

source *ANALYTICS_TOP*/conf/cshrc.perf

- For **sh**, **ksh**, or **bash**:

  . *ANALYTICS_TOP*/conf/profile.perf

Table 3. Actions on the Analytics node data sources

Action	Command line
Add a node data source	UNIX: `dbconfig.sh 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`
Edit the settings of any node data source, including FLEXnet Manager data sources	UNIX: `dbconfig.sh edit` *data_source_name* where: • *data_source_name* is the name the data source that you want to edit.

Chapter 3. Managing the Analytics node

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

Loader controllers

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

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`

Default loader controller 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.

Configuration to modify loader controller behavior

Table 4 on page 10 lists the configuration action needed to modify the behavior of the loader controller.

Table 4. Configuration action 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`	`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

You can perform actions to view, start, and stop the loader controller service, and to change the loader controller settings.

Actions on the loader controller service

Table 5 lists the actions you can perform 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.

Table 5. Actions on the loader controller 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

Table 6 lists the action you can perform to change the loader controller settings.

Table 6. Action 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* where: • *log_level* is the log level of your loader controller log file. If you restart the loader controller, these settings will revert to the default level. **Note:** You must run this command on the local host running the **plc** service.

Data loaders

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

Data loaders are polling loaders or history data loaders. The data loaders gather data and load it into specific tables in the relational database as raw data.

Normally, data loaders perform synchronous data loading, whereby they load data directly into the Analytics database. In rare cases where the network connection between the Analytics node and the database host is poor, the data loaders will perform asynchronous data loading. In such cases, the data loaders send data to the Analytics server, and the server then loads the data into the Analytics database.

Data loaders automatically handle daylight saving time 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 and 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`

Default data loader 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`)

Table 7 lists the LSF host data loaders.

Table 7. LSF host data loaders

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`)

Table 8 lists the LSF job data loaders.

Table 8. LSF job data loaders

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`)

Table 9 lists the LSF data loaders.

Table 9. LSF data loaders

Data loader name	Data type	Data gathering interval	Data loads to	Loader type
Host metrics (**hostmetricsloader**)	host-related metrics	10 minutes	RESOURCE_METRICS_BUILTIN RESOURCE_METRICS_ELIM	polling
Host properties (**hostpropertiesloader**)	resource properties	1 hour	RESOURCE_ATTRIBUTES HOST_BOOLEANRES	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, **JOB_FINISH2** events	5 minutes	LSB_EVENTS LSB_EVENTS_EXECHOSTLIST LSF_PERFORMANCE_METRIC LSB_JOB_FINISH LSB_JOB_EXECHOSTS LSB_JOB_STARTLIMIT	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`)

Table 10 lists the LSF advanced data loaders.

Table 10. LSF advanced data loaders

Data loader name	Data type	Data gathering interval	Data loads to	Loader type
Host group (**hostgrouploader**)	host group	1 hour	HOST_GROUP	polling
Bqueues (**lsfbqueueloader**)	queue properties	5 minutes	LSF_BQUEUES	polling
Pending reason (**lsfpendingreasonloader**)	job pending reasons	15 minutes	JOBS_PENDING_REASON DPR_BYINTERVAL	polling
User group (**usergrouploader**)	user group	1 hour	USER_GROUP	polling
Pending Reasons (**lsbpendingreasonsloader**)	job pending reason - from the LSF data file `lsb.pendingreasons`	10 minutes	LSB_JOB_PENDINGREASON	file
Job status (**lsfjobstatusloader**)	job status - from the LSF data file `lsb.status`	10 minutes	LSB_JOB_STATUS	file

FLEXnet data loaders (`plc_license.xml`)

Table 11 lists the FLEXnet data loaders.

Table 11. FLEXnet data loaders

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

Data loader interactions

The loader controller service controls the scheduling of the data loaders. The data loaders store Platform LSF data 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.

Figure 5 on page 14 illustrates the interaction between the data loaders and other components.

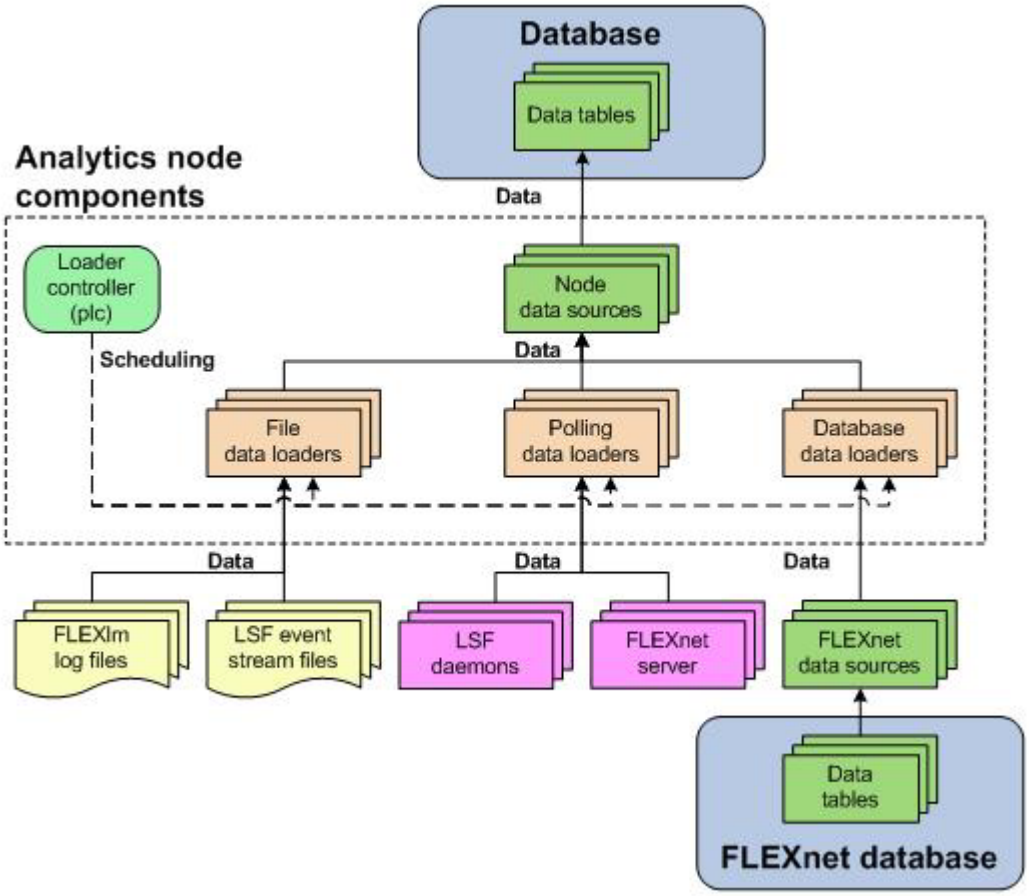

Figure 5. Interaction between data loaders and other components

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`

Table 12. Configuration actions to modify data loader behavior

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: `$PERF_CONFDIR/dataloader`	`<Writer ... EnableRecover="Y">`
Disable data loss protection for the specified data loader.		`<Writer ... EnableRecover="N">`
Specify the default log level of your data loader log files.	`log4j.properties` File location: UNIX: `$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

Table 13 lists the actions you can perform on the data loaders.

Table 13. Data loader actions

Action	Command line
View the status and logging levels of the data loaders.	UNIX: `plcclient.sh -s`

Table 13. Data loader actions (continued)

Action	Command line
Dynamically change the log level of your data loader log files (temporarily).	UNIX: `plcclient.sh -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 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` 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 to the default level.

Viewing or dynamically editing the data loader settings

Use the Platform 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).

Procedure

1. In the navigation tree of the Platform 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**.

Analytics node command-line tools

dbconfig

Use the **dbconfig** command to configure the node data source.

Synopsis

UNIX commands:

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

dbconfig.sh -h

Description

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

If you are running this command locally on 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.

If the Analytics node is running on a UNIX host, you must source the Analytics environment before running the **dbconfig.sh** command.

- For **csh** or **tcsh**:

  `source ANALYTICS_TOP/conf/cshrc.perf`
- For **sh**, **ksh**, or **bash**:

  `. ANALYTICS_TOP/conf/profile.perf`

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

Use the **perfadmin** command to 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

Use the **plcclient** command to 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*]

Description

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

Options

-s View the status of the data loaders

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

-n *loader_name* **-l** *log_level*
Dynamically change the log level of the specified data loader to the specified log level. If you restart the loader controller (**plc**) service, this setting will revert back to the default level.

Analytics node configuration files

The configuration files for the Analytics node are:

- "perf.conf"

perf.conf

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

About perf.conf

The `perf.conf` file 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 Platform Analytics and applications built on top of it. For example, information in `perf.conf` is used by Platform 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 Platform 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 the `/conf` directory. 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 (`#`) is a comment and is 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 Platform 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
    ```
- 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:

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

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

Multiple files may be kept in the same directory, but each one must reference a different license server. When checking out a license, Platform 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, Platform 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 Platform Analytics installation.

If you installed FLEXlm separately from Platform 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`

LICENSE_VERSION

Syntax

LICENSE_VERSION=*version_number*

Description

Specifies the version of the license module installed with Platform 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`

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

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*

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

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

Chapter 4. Managing the Analytics server

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

The server performs the following functions:

- Analytics node management
- Cluster data management

Platform Analytics Console

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

Clusters

Displays information about each cluster that Platform 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 Platform Analytics. You can filter the display of these events to find specific events.

Platform Analytics Console actions

Table 14 lists the actions you can take on the Platform Analytics Console.

Table 14. Platform Analytics Console actions

Action	Command line
Start the Platform Analytics Console.	• Windows: **Start** > **All Programs** > **>IBM Corporation** > **Analytics Server** > **Analytics Console** **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 Platform Analytics Console will not start up until you can connect to the data source.

Data transformers

At regular intervals, data transformers convert raw cluster data in the Analytics database into a format usable for reporting and analysis.

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:

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

Default data transformer behavior

Data transformers convert data at regular 10-minute intervals. Table 15 lists the data transformers and the database tables in which the data transformers generate the data.

Table 15. Data transformers and transformed database tables

Data transformer name	Transformed database tables
ClusterCapacity	RPT_CLUSTER_CAPACITY_RAW
FlexLMLicusage	RPT_FLEXLM_LICUSAGE_RAW
FNMLicusage	RPT_FNM_LICUSAGE_RAW RPT_FNM_LICUSAGE_BY_FEATURE RPT_FNM_LICUSAGE_BY_SERVER
FNMWorkload	RPT_FNM_WORKLOAD_RAW
Hardware	RPT_HARDWARE_RAW RPT_HARDWARE_DAY
JobPendingReason	RPT_JOB_PENDINGREASON_RAW
LicenseDenials	RPT_LICENSE_DENIALS_RAW
WorkloadAccounting and Resource Usage	RPT_JOBMART_RAW RPT_JOBMART_DAY
WorkloadStatistics	RPT_WORKLOAD_STATISTICS_RAW

See Appendix A, “Database report table (RPT) descriptions,” on page 65 for complete descriptions of these database tables.

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.

Figure 6 on page 27 illustrates the interaction between the data transformers and other components.

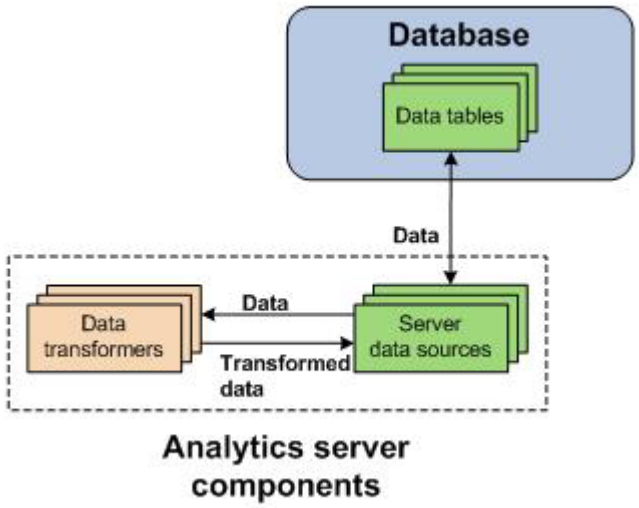

Figure 6. Data transformer interaction with other components

Configuration to modify data transformer behavior

Table 16 lists the configuration actions you can perform to modify data transformer behavior.

Table 16. Configuration actions 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 30).

Event notification

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

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

Event notifications

Platform 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 Platform Analytics or the cluster, allowing you to decide whether you want to check the Platform Analytics Console for further details.

Event actions

Table 17 lists the actions you can take on events and event notifications.

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.

Table 17. Event and event notification actions

Action	Platform 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 "Restarting the Platform Task Scheduler."

Restarting the Platform Task Scheduler

If you enable or disable event notification, you need to restart the Platform Task Scheduler to apply this change.

Procedure

For an Analytics server running on a Windows host: Restart the task scheduler service.

1. From the Windows Control Panel, select **Administrative Tools > Services**.
2. Right-click **Analytics Task Scheduler** and select **Restart**.

Configuration to modify event notification behavior

Table 18 lists the configuration actions you can perform to modify event notification behavior.

Table 18. Configuration actions 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.

Data purger

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

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.

Figure 7 illustrates the interaction between the data purger and other components.

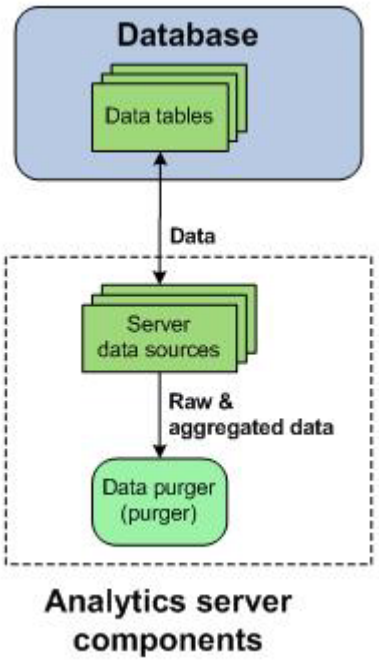

Figure 7. Interaction between data purger and other components

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").

Scheduled tasks

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

After metric data is collected from hosts and stored in the database, the data undergoes several processing tasks for maintenance purposes. Platform 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

Platform 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

Platform 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.

Daily report (`DailyReportETL`)

The daily report scheduled task builds jobmart data to the RPT_JOBMART_DAY table and hardware data to the RPT_HARDWARE_DAY table. By default, this task runs every day.

Cluster and workload (`HostRelatedETL`)

The cluster and workload scheduled task builds jobmart data to the RPT_CLUSTER_CAPACITY_RAW table and hardware data to the RPT_HARDWARE_RAW table. By default, this task runs every hour.

Hardware jobmart (`JOBRelatedETL`)

The hardware jobmart scheduled task builds jobmart data to the RPT_JOBMART_RAW table, workload statistics data to the RPT_WORKLOAD_STATISTICS_RAW table, and pending reason data to the RPT_JOB_PENDINGREASON_RAW table. By default, this task runs every hour.

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 29.

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 Analytics 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

Table 19 lists the actions you can take on scheduled tasks.

Table 19. Scheduled task actions

Action	Platform 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 "Creating, editing, or viewing a scheduled task" for detailed information.
View or edit a task from the list of scheduled tasks.	See "Creating, editing, or viewing a scheduled task" 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**.

Creating, editing, or viewing a scheduled task

Perform this task to create, edit, or view a scheduled task.

About this 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

Procedure

1. In the navigation tree of the Platform 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.

Attention: Do not change the name of the scheduled task; otherwise, Platform 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:

   - 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 Platform 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**.

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 Platform Analytics console.

Synopsis

Windows command:

runconsole

Analytics server configuration files

- "pi.conf"

pi.conf

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

About pi.conf

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

Changing pi.conf configuration

After making any changes to the `pi.conf` file, run the following commands from the *ANALYTICS_TOP*`/bin` directory to restart the Platform Analytics server and apply your changes:

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

Location

The location of `pi.conf` is in the *ANALYTICS_TOP*`/conf` directory.

Format

Each entry in the `pi.conf` file 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 Platform Analytics Console (in the navigation tree, click **Events**, then right-click on the list of events and select **Action** > **Notification**).

If set to true, Platform 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 Platform Analytics or the cluster, allowing you to decide whether you want to check the Platform Analytics Console for further details.

For more information on event notification, refer to "Event notification" on page 28.

Default

`true`

mail.smtp.host
Syntax

mail.smtp.host=*host_name.domain_name*

Description

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

You would normally configure this parameter using the Platform 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 Platform Analytics uses to send event notification emails.

You would normally configure this parameter using the Platform 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 Platform Analytics will send.

You would normally configure this parameter using the Platform 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 Platform Analytics will send.

You would normally configure this parameter using the Platform 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 Platform 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 Platform 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 Platform Analytics Console to display the following optional columns:

- System Purpose
- Display Description
- Business Area

Default

`YES`

Chapter 5. Platform Analytics reports

The support hosts, such as the Platform Analytics reporting server, Platform Analytics Designer, and Platform Application Center, do not run Platform Analytics. They are necessary in order for you to take full advantage of the cluster operations data and reports that Platform Analytics assembles and generates.

Generating reports

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

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

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

Table 20 lists the default workbooks provided by the Platform Analytics reporting server to allow you to analyze your clusters.

Table 20. Default workbooks provided by the Platform Analytics reporting server

Workbook name	Description
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.
FlexLM Denials	Reports FlexLM Server denial events and license denials on any license server or across multiple license servers.
FlexLM License Usage	Reports 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.
FNM Denials	Reports FlexNet Manager (FNM) denial events and license denials on any license server or across multiple license servers.
FNM License Usage	Reports FlexNet Manager (FNM) license usage on any license server or across multiple license servers. This allows you to analyze the usage, consumption, and utilization of licenses by features and servers.
FNM Workload Accounting	Reports license usage for jobs that use licenses.
Hardware	Reports hardware utilization at any time period.
Pending Reasons	Reports the number of pending reason instances for different reasons at any period in time.
Resource Memory Requested Vs Used	Reports wasted memory usage information by comparing requested and used memories.
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.

Table 20. Default workbooks provided by the Platform Analytics reporting server (continued)

Workbook name	Description
Workload Accounting (Daily) and Hardware (Daily)	Data is aggregated daily for better workbook performance.
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.

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

Reporting server interactions

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

Figure 8 illustrates the interaction between the reporting server and other components.

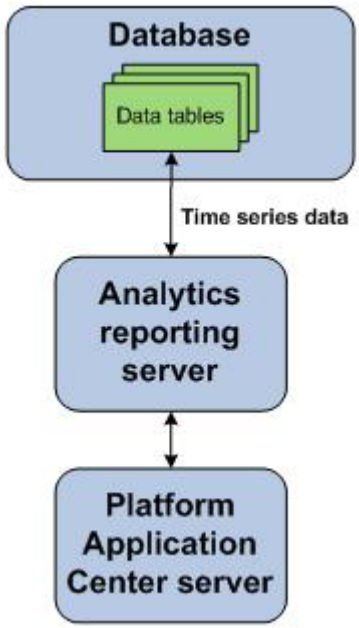

Figure 8. Platform Analytics reporting server interactions

Collecting data and viewing reports

The Platform Analytics reporting server generates Platform Analytics reports and allows other users to view these reports. In order to view reports, you need to first collect data, publish them to Analytics reporting server, and view them.

Collecting data

If you want to collect FLEXlm usage and FLEXlm events data, start the license servers and configure the Analytics node.

Procedure

1. Start the LSF cluster.

   Run **lsfstartup** after sourcing the lsf.profile file.

2. Start the license server daemon.

   a. Log on to the license server host as LSF administrator.

   b. Run the lmgrd command in LSF_SERVERDIR to start the license server daemon:

   ```
   lmgrd -c /usr/share/lsf/lsf_62/conf/license.dat -1/usr/share/lsf/lsf_62/log/license.log
   ```

   c. Make sure that the FLEXnet data loaders are enabled in your cluster.

3. Start the database.

   a. Open the **Administration Tools**.

   b. On the **Main Menu**, select **Start Database**.

4. Start the Platform Analytics node and source LSF and perf environment.

   ```
   perfadmin start plc | all
   plcclient [-s]
   ```

   Check the plc configuration file for any errors plc.log.<*host_name*> under the ANALYTICS_TOP/log directory.

   Check log file of individual loaders (<*dataloader_name*>.log.<*host_name*>) under the ANALYTICS_TOP/log/dataloader directory for details of individual data loaders.

   You can even check the database table to see if data has been successfully loaded into the database.

5. Start the Platform Analytics server and transform data

   ```
   perfadmin start all
   runconsole
   ```

   Check log files under the ANALYTICS_TOP/log directory for details.

Viewing reports

Once data is collected in the database, you can view reports using the Analytics reporting server. Optionally, you can even view reports using Platform Analytics Designer or Platform Application Center.

Procedure

1. Log in to the Platform Analytics reporting server.

   http://*host_name*:*port*

   where *host_name* is the name of the system where Tableau Server is installed and *port* is the number which you entered during the Tableau Server installation.

2. You can view workbooks, worksheets, and dashboards.

   Workbook
   A Tableau Server report (twb) file. It consists of dashboards and worksheets.

   Dashboard
   A view of multiple worksheets.

   Worksheet
   A single view of queried data from a data source. This may be a table or a chart. A worksheet does not have to be viewed via a dashboard, it can be accessed directly, if required.

Platform Application Center (optional)

Platform Application Center embeds IBM Platform Analytics. You must install the Platform Application Center Analytics add-on package to avail advanced web-based analysis and reporting on LSF data. The package comes with installation instructions. You can download the add-on package from the same location as Platform Analytics.

With the integration of Platform Analytics and Platform Application Center you can:

- Schedule and monitor jobs
- Subscribe to a report, or unsubscribe from a report to receive email messages when reports are updated
- Add extra email addresses for sending reports
- View past reports

For more details, see the Platform Application Center documentation.

Platform Application Center host interactions

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

Figure 9 illustrates the interaction between the support hosts and other components.

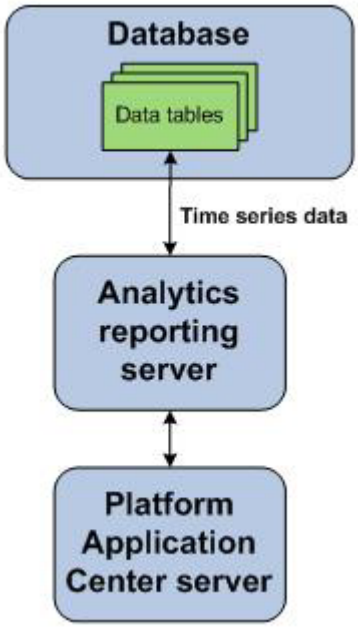

Figure 9. Platform Application Center host interactions

About HTTPS

Configuring HTTPS is optional.

You can configure HTTPS on both IBM Platform Application Center and Tableau Server using a self-signed certificate.

You can configure HTTPS only for Platform Application Center, only for Tableau Server, or for both.

When you configure HTTPS for Platform Application Center, it affects access to the web server (URL will start with `https:`), access to Web Services, and the Report Builder (Report Builder will need a certificate to communicate with Platform Application Center).

When you configure HTTPS for Tableau Server, it affects report generation and workbook access.

Chapter 6. Managing Platform Analytics

Managing Platform Analytics includes:

- "Securing your data and working environment"
- "Maintaining the Analytics database" on page 48
- "Troubleshooting the Analytics node" on page 50
- "Troubleshooting the Analytics server" on page 56

Securing 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

- "Opening ports to communicate across firewalls"
- "Modifying the database password" on page 48

Opening 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.

About this task

Platform Analytics uses the ports listed in Table 21 to communicate with other hosts in the cluster:

Table 21. Platform Analytics ports

Port name	Default port number	Additional information
PIEM_PORT	9091	Internal port for the event manager. Used for receiving events from Platform Analytics components. Configuration is not required,
PIAM_PORT	9092	Internal port for the automation manager. Used for receiving events from Platform Analytics components. Configuration is not required.
Remoting server port (asynchronous data loading mode only)	9093	Internal port for the remoting server. Used for communicating 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.

Procedure

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

Modifying the database password

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

Procedure

1. Log in to the Platform 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**.

Maintaining the Analytics database

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

Actions to maintain the Analytics database

- Partitioning tables in the database

  You can partition data tables in the Analytics database, which divides 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 details on recovering the database, see "Partitioning Tables" in the Vertica *Administrator's Guide*.
- Recovering the database

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

  For details on recovering the database, see "Recovering the Database" in the Vertica *Administrator's Guide*.
- Backing up or restoring 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.

Backing up and restoring data in the database

You can back up or restore data in the database using full backup or incremental backup scripts.

Always back up the data in the database before performing any of these tasks:

- Upgrading to a newer version of the database software
- Dropping a partition
- Adding a node to the database cluster

Attention: Observe the following important points before backing up your data:

- Make sure you have installed rsync 3.0 or later on the database nodes. You can use the **rsync --version** command to check the version.
- Check the disk space on every database node and make sure that the backup directory has enough space.
- The **backup.sh** script works only if the database is up and running. You can use **admin** tools in Vertica to check the database status.
- Note the snapshot name used by the **backup.sh** script for later use in restore operations.
- Perform a full backup at least once a week and an incremental backup every other day.

Full backup

You can either use cold backup or hot backup to back up all the data on the drive.

- Cold backup

  This is an offline backup. Make sure that the database is down before you copy all data to a backup directory.
- Hot backup

  This is a dynamic backup. Vertica provides a utility to perform a full backup called **backup.sh**.

  For more information about backing up or restoring data in the database, see "Backup and Restore" in the Vertica *Administrator's Guide*.

Incremental Backup

You can do an incremental backup to back up data that has changed or is new since the last incremental backup. This method takes less time to back up data compared to a full backup.

1. Do a hot backup first. Vertica creates a snapshot file. This file is found in the location where you set the **-B** parameter when you used **backup.sh** to do a full back up your database.
2. Do the incremental backup. You can also use **backup.sh** (in `$vertica_top/scripts/`) to do this. You must specify the snapshot file that was created by the full backup. For details, see "Backup and Restore" in the Vertica *Administrator's Guide*.

   You can write a script to run an incremental backup every other day. For example:

   ```
   /opt/Vertica/scripts/backup.sh –s host1,host2,host3 –i host1 –b host1 –B
   /backupDir –D /vdata/pa8 –d pa8 –u dbadmin –w dbadmin –S backup1
   ```

   This creates a backup from a three-node system and is run from host1, initialized by host1, with the backup stored under `/backupDir`.

Restore

Attention: Observe the following important points before restoring your data:

- The backup must have been created using the **backup.sh** script. Note the snapshot name used by the **backup.sh** script for use in restore operations.
- By default, **restore.sh** does not restore the `vertica.conf` file. This is useful if you have modified the database configuration since the database was backed up. Use the **restore.sh** script with the **-c** option to restore the `vertica.conf` file. For example:

  ```
  restore.sh -c
  ```
- Make sure to shut down the database before running the restore script.

Use the **restore.sh** script (in `/opt/vertica/scripts/`) to restore the database from a backup created by the **backup.sh** script. For example:

```
/opt/Vertica/bin/restore.sh –s host1,host2,host3 –b host1 –B /backupDir –D /vdata/pa8 –S backup1
```

This restores snapshot `backup1` to a three-node system from backup directory `/backupDir` from backup host, host1.

Troubleshooting the Analytics node

Perform these tasks to troubleshoot problems with the Analytics node.

Changing 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.

Procedure

1. If you are logged in to 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`

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).

Disabling data collection for individual data loaders

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

Procedure

1. If you are logged in to 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`
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.

Checking the status of the loader controller

Perform this task to check the status of the loader controller.

Procedure

1. If you are logged in to 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 $PERF_TOP/version_number/bin`
4. View the status of the loader controller (**plc**) and other PERF services.
   `perfadmin list`
5. Verify that there are no errors in the loader controller log file.
   The loader controller log file is located in the log directory:
   - UNIX: `$PERF_LOGDIR`

Checking the status of the data loaders

Perform this task to check the status of the data loaders.

Procedure

1. If you are logged in to 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 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`

Checking the status of the Analytics node database connection

Perform this task to check the status of the Analytics node database connection.

Procedure

1. If you are logged in to 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 binary directory.
   - UNIX: `cd $PERF_TOP/version_number/bin`
4. View the status of the node database connection.
   - UNIX: `dbconfig.sh`

Checking core dump on the Analytics node

Perform these tasks, depending on your operating system, to check and enable core dumps.

Core dump on Linux

Perform this task to check and enable core dumps on Linux systems.

Procedure

1. If you are logged in to 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. 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

Perform this task to check and enable core dumps on Solaris systems.

Procedure

1. If you are logged in to 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. 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

Perform this task to check and enable core dumps on AIX® and HP-UX systems.

Procedure

1. If you are logged in to 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. 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`

Debugging the LSF API

Perform this task to enable debugging for the LSF API.

Procedure

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 is not responding

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.

Procedure

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 32-bit systems, the maximum heap size that you can set is 1600M. For Linux / Unix 32bit, you can set it to 4096M. For 64-bit systems, 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 must restart the loader controller once you increase the disk space.

Troubleshooting the Analytics server

Perform these tasks to troubleshoot problems with the Analytics server.

- "Checking the health of the Analytics server" on page 57
- "Checking the Analytics server log files" on page 57
- "Checking the status of the Analytics server database connection" on page 57

Checking the health of the Analytics server

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

Procedure

1. Log in to the Analytics server.
2. Launch the Platform Analytics Console.
   - Windows: **Start** > **All Programs** > **>IBM Corporation** > **Analytics Server** > **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.

Checking the Analytics server log files

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

Procedure

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:
   - 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:
   - 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:
   - Windows: *`ANALYTICS_TOP`*`\log`

Checking the status of the Analytics server database connection

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

Procedure

1. Log in to the Analytics server host.
2. Launch the Platform Analytics Console.

- Windows: **Start** > **All Programs** > **>IBM Corporation** > **Analytics Server** > **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.

Chapter 7. Customizing Platform Analytics

Platform Analytics customizations allow you to maintain and upgrade your Platform Analytics installation to improve performance and fix issues.

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

Naming conventions

The name of the customization is the same as the package name and identifies the specific customization, allowing us 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.

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 topics describe conventions and examples of customizations to the Analytics node:

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 Platform 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 Platform Analytics files not in this supported list.

Customizing an existing data loader

This task describes how to customize an existing data loader.

About this task

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

Procedure

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.

Example

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 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.

Adding a new custom data loader

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

Procedure

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.

Example

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 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 topics describe conventions and examples of customizations to the Analytics server:

- "Supported files"
- "Customizing an existing workbook"

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 Platform 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 Platform Analytics files not in this supported list.

Customizing an existing 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 Platform Analytics Designer 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 actions 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 tasks describe the conventions while assembling, installing, or viewing the customization packages (or "patches").

- "Assembling the customization package"
- "Installing the customization package" on page 64
- "Viewing details on the customization packages" on page 64

Assembling the customization package

Perform this task to assemble the customization package.

About this task

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 Platform Analytics patch installer, and add the following text files to this subdirectory:

Procedure

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 that details patch information in the following format:

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

      where:

      build_number
      The build request number. This build number 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 naming conventions. For example, `12345`.

      build_date
      The build date in `UTC/GMT` time in the following numerical format: `YYYYMMDDhhmmss`. For example, `20111015104104`.

      version
      The version of your Platform Analytics installation. For example, `9.1`.

      dependency
      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 dependencies, use `null`. For example, `1234,2345`.

      manual_config
      Specifies whether the patch has manual configuration steps before starting the Platform Analytics services. If set to `Y`, the patch installer does not restart Platform Analytics services after deploying the patch; otherwise, the patch installer will restart the Platform Analytics services after deploying the patch. The default value is `N`.

For example:

```
12345;20111015104104;9.1;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 of 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
```

Installing the customization package

Perform this task to install the customization package.

Procedure

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

Notes:

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

Viewing details on the customization packages

The following commands allow you to view information about the customization that are applied to the Platform Analytics installation.

Procedure

- List information on all patches applied to the current Platform 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_number*
  - Windows: `pversion.bat -b` *build_number*

Appendix A. Database report table (RPT) descriptions

If you are planning to customize reports, then it is necessary for you to understand the report tables.

RPT_HARDWARE_RAW

Table 22. RPT_HARDWARE_RAW. This table stores raw hardware data for reporting.

Column name	Data type	PK	Description
CLUSTER_NAME	VARCHAR(128)	Y	The name of the LSF cluster.
TIME_STAMP	TIME_STAMP	Y	The time that the sample is taken.
ISO_WEEK	VARCHAR (10)		In the format `TO_CHAR(TIME_STAMP, 'IYYY-IW')`.
TIME_STAMP_GMT	NUMBER(13)		Event expected log time in GMT time zone, presented as the number of seconds after 1970/01/01.
HOST_NAME	VARCHAR(128)	Y	The name of the host in the cluster.
CLUSTER_HOST	VARCHAR(257)		The concatenation of CLUSTER_NAME and HOST_NAME.
LSFHOST_TYPE	VARCHAR(128)		Type of host you have, such as LINUX86.
LSFHOST_MODEL	VARCHAR(128)		The host model of the host, such as UltraSparc10.
CPU_FACTOR	NUMBER(10,4)		Speed of the host's CPU relative to other hosts in the cluster. If one processor is twice the speed of another, its CPU factor should be twice as large. The CPU factors are defined by the administrator. For multiprocessor hosts, the CPU factor is the speed of a single processor; the system automatically scales the host CPU load to account for additional processors.
NCPUS	NUMBER(10,4)		Number of CPUs you have specified for your host.
NPROCS	NUMBER(10,4)		Number of physical processors (if NCPUS is defined as procs, then NCPUS = NPROCS)
NCORES	NUMBER(10,4)		Number of cores per processor (if NCPUS is defined as cores, then NCPUS = NPROCS × NCORES).
NTHREADS	NUMBER(10,4)		Number of cores per processor (if NCPUS is defined as cores, then NCPUS = NPROCS × NCORES).
HOST_GROUP	VARCHAR(128)		The user defined LSF HOST_GROUP that the host belongs to.
HOST_STATUS	VARCHAR(64)		LSF Status of the host. Can be `OK`, `Closed_Excl`, `Unreach`, `Closed_Full`, `Closed_Busy`, and so on.
MAX_SLOT	NUMBER(19,4)		Maximum slots that this host has.
RUN_SLOT	NUMBER(19,4)		The number of slots that have running jobs.
LS	NUMBER(19,4)		Number of current users logged on to the system.
IT	NUMBER(19,4)		Amount of time in minutes that a host has been idle. On a Linux/UNIX host, it is the amount of time since the keyboard has been touched on all logged in sessions. On a Windows host, it is the amount of time a screen saver has been active

Table 22. RPT_HARDWARE_RAW (continued). This table stores raw hardware data for reporting.

Column name	Data type	PK	Description
R15M	NUMBER(19,4)		Load this host carries, averaged over the last 15 minutes. The load is the average number of processes using the CPU during a given time interval.
R15S	NUMBER(19,4)		Load this host carries, averaged over the last 15 seconds. The load is the average number of processes using the CPU during a given time interval.
R1M	NUMBER(19,4)		Load this host carries, averaged over the last minute. The load is the average number of processes using the CPU during a given time interval.
UT	NUMBER(19,4)		Current CPU utilization of your host, as a percentage.
IO	NUMBER(19,4)		I/O throughput to disks attached directly to this host, in KB per second. This rate does not include I/O to disks that are mounted from other hosts.
MEM	NUMBER(19,4)		Estimate of the real memory, in MB, currently available to user processes. This represents the approximate size of the largest process that could be started on a host without causing the host to start paging.
SWP	NUMBER(19,4)		Currently available virtual memory (swap space) in MB. This represents the largest process that can be started on the host (with paging).
TMP	NUMBER(19,4)		Space available in MB on the file system that contains the temporary directory.
MAX_MEM	NUMBER(19,4)		Maximum RAM available.
MAX_SWP	NUMBER(19,4)		Maximum swap space on your host.
MAX_TMP	NUMBER(19,4)		Maximum space in `/tmp` (Linux/UNIX) or OS default temp directory (Windows).
PG	NUMBER(19,4)		Virtual memory paging rate in pages per second. This index is closely tied to the amount of available memory and the total size of the processes running on a host; if there is not enough memory to satisfy all processes, the paging rate is high.
RESOURCE_METRICS_INTERVAL	NUMBER(19,4)		The sampling interval of resource_metrics loader. This is for aligning the resource_metrics and bhosts sampling. Usually is 10 minutes.
LSF_BHOSTS_INTERVAL	NUMBER(19,4)		Bhosts loader sampling interval. Usually is 10 minutes.
CLUSTER_MAPPING	VARCHAR(4000)		This is an unused column for user/PS to add a mapping for Cluster name such as mapping cluster to a business unit.

RPT_HARDWARE_DAY

Table 23. RPT_HARDWARE_DAY. This table stores hardware data, aggregated to the daily level.

Column name	Data type	PK	Description
CLUSTER_NAME	VARCHAR(128)	Y	The name of the LSF cluster.
TIME_STAMP	TIMESTAMP	Y	The time that this sample is taken.
ISO_WEEK	VARCHAR (10)		In the format `to_char(TIME_STAMP, 'IYYY-IW')`

Table 23. RPT_HARDWARE_DAY (continued). This table stores hardware data, aggregated to the daily level.

Column name	Data type	PK	Description
TIME_STAMP_GMT	NUMBER(13)	Y	Event expected log time in GMT time zone, presented as the number of seconds after 1970/01/01.
HOST_NAME	VARCHAR(128)	Y	The name of the host in this cluster.
CLUSTER_HOST	VARCHAR(257)	Y	Append CLUSTER_NAME and HOST_NAME together.
LSFHOST_TYPE	VARCHAR(128)	Y	Type of host you have. For example, LINUX86.
LSFHOST_MODEL	VARCHAR(128)	Y	The host model of this host, such as UltraSparc10.
CPU_FACTOR	NUMBER(10,4)	Y	Speed of the host's CPU relative to other hosts in the cluster. If one processor is twice the speed of another, its CPU factor should be twice as large. The CPU factors are defined by the administrator. For multiprocessor hosts, the CPU factor is the speed of a single processor; the system automatically scales the host CPU load to account for additional processors.
NCPUS	NUMBER(10,4)	Y	Number of CPUs you have specified for your host.
NPROCS	NUMBER(10,4)	Y	Number of physical processors (if NCPUS is defined as procs, then NCPUS = NPROCS) .
NCORES	NUMBER(10,4)	Y	Number of cores per processor (if NCPUS is defined as cores, then NCPUS = NPROCS × NCORES).
NTHREADS	NUMBER(10,4)	Y	Number of threads per core (if NCPUS is defined as threads, then NCPUS = NPROCS × NCORES × NTHREADS).
HOST_GROUP	VARCHAR(128)	Y	The user defined LSF HOST_GROUP this host belongs to.
HOST_STATUS	VARCHAR(64)	Y	LSF status of the host. Could be `OK`, `Closed_Excl`, `Unreach`, `Closed_Full`, `Closed_Busy`, and so on.
MAX_SLOT	NUMBER(19,4)		Maximum slots that this host has.
RUN_SLOT	NUMBER(19,4)		The number of slots that are running jobs.
LS	NUMBER(19,4)		Number of current users logged in to the system.
IT	NUMBER(19,4)		Amount of time in minutes that a host has been idle. On a Linux/UNIX host, it is the amount of time since the keyboard has been touched on all logged in sessions. On a Windows host, it is the amount of time a screen saver has been active.
R15M	NUMBER(19,4)		Load this host carries, averaged over the last 15 minutes. The load is the average number of processes using the CPU during a given time interval.
R15S	NUMBER(19,4)		Load this host carries, averaged over the last 15 seconds. The load is the average number of processes using the CPU during a given time interval.
R1M	NUMBER(19,4)		Load this host carries, averaged over the last minute. The load is the average number of processes using the CPU during a given time interval.
UT	NUMBER(19,4)		Current CPU utilization of your host, as a percentage.
IO	NUMBER(19,4)		I/O throughput to disks attached directly to this host, in KB per second. This rate does not include I/O to disks that are mounted from other hosts.

Table 23. RPT_HARDWARE_DAY (continued). This table stores hardware data, aggregated to the daily level.

Column name	Data type	PK	Description
MEM	NUMBER(19,4)		Estimate of the real memory, in MB, currently available to user processes. This represents the approximate size of the largest process that could be started on a host without causing the host to start paging.
SWP	NUMBER(19,4)		Currently available virtual memory (swap space) in MB. This represents the largest process that can be started on the host (with paging).
TMP	NUMBER(19,4)		Space available in MB on the file system that contains the temporary directory
MAX_MEM	NUMBER(19,4)		Maximum RAM available.
MAX_SWP	NUMBER(19,4)		Maximum swap space on your host.
MAX_TMP	NUMBER(19,4)		Maximum space in `/tmp` (Linux/UNIX) or OS default temp directory (Windows).
PG	NUMBER(19,4)		Virtual memory paging rate in pages per second. This index is closely tied to the amount of available memory and the total size of the processes running on a host; if there is not enough memory to satisfy all processes, the paging rate is high.
RESOURCE_METRICS_INTERVAL	NUMBER(19,4)		The sampling interval of resource_metrics loader. This is for aligning the resource_metrics and bhosts sampling. Usually is 10 minutes.
LSF_BHOSTS_INTERVAL	NUMBER(19,4)		Bhosts loader sampling interval. Usually is 10 minutes.
SAMPLING_COUNT	NUMBER(19,4)		The number of records that have been aggregated into this one.
CLUSTER_MAPPING	VARCHAR(4000)		This is an unused column for user/PS to add a mapping for Cluster name such as mapping cluster to a business unit.

RPT_CLUSTER_CAPACITY_RAW

Table 24. RPT_CLUSTER_CAPACITY_RAW. This table is used for the cluster capacity report. The data comes from lsf_bhosts and lsf_bjobs aggregated to the hourly level.

Column name	Data type	PK	Description
CLUSTER_NAME	VARCHAR(128)	Y	The name of the LSF cluster.
TIME_STAMP	TIME_STAMP	Y	The time that this sample is taken.
TIME_STAMP_GMT	TIMESTAMP		Event expected log time in GMT time zone, presented as the number of seconds after 1970/01/01.
ISO_WEEK	VARCHAR(10)		In the format `to_char(TIME_STAMP, 'IYYY-IW')`.
CATEGORY	VARCHAR(64)	Y	This column identifies the job state a slot is in, such as `RUN`, `IDLE`, `CLOSED`, `DOWN`, or `UNUSEDEXCLUSIVE`.
HOST_NAME	VARCHAR(512)	Y	The name of the host in the cluster.
USER_NAME	VARCHAR(128)	Y	The user name that is running the job on that host (if CATEGORY = 'RUN'; otherwise, it is `'-'`).
QUEUE_NAME	VARCHAR(128)	Y	The queue name on which the job is running; otherwise, it is `'-'`.
PROJECT_NAME	VARCHAR(4000)	Y	The project name under which the job is running; otherwise, it is `'-'`.
JOB_GROUP	VARCHAR(4000)	Y	The job_group in which the job is running; otherwise, it is `'-'`.

Table 24. RPT_CLUSTER_CAPACITY_RAW (continued). This table is used for the cluster capacity report. The data comes from lsf_bhosts and lsf_bjobs aggregated to the hourly level.

Column name	Data type	PK	Description
USER_GROUP	VARCHAR(512)	Y	The user_group in which the job is running; otherwise, it is '-'.
HOST_TYPE	VARCHAR(128)		The LSF host_type to which the host belongs, such as Linux86.
HOST_MODEL	VARCHAR(128)		The LSF host_model to which the host belongs, such as UltraSparc10.
HOST_GROUP	VARCHAR(128)		The user-defined host_group to which this host belongs.
SLOTS	NUMBER(19,4)		The sum of slots aggregated by all jobs that ran during this time sample on this host, user, job_group, project, queue, host_group, user_group, host_type and host_model. For other status, it is the number of slots for that status on this host in this sample time.
MEM_USAGE	NUMBER(19,4)		The sum of max memory used by jobs in this time sample with the same host, user, job_group, project, queue, host_group, user_group, host_type and host_model. For other status, this column will be null.
CPU_DELTA	NUMBER(19,4)		The sum of cpu_delta for all jobs in this sample with the same host, user, job_group, project, queue, host_group, user_group, host_type and host_model. For other status, this column will be 0 or null.
SLOTS_HOUR	NUMBER(19,4)		This is the sum of the product of the number of slots used times the sampling time interval for the hour for that category status.
CLUSTER_MAPPING	VARCHAR(4000)		Reserved column for mapping a cluster name to a customization value, such as department.
PROJECT_MAPPING	VARCHAR(4000)		Reserved column for mapping a project name to a customization value, such as department.
USER_MAPPING	VARCHAR(4000)		Reserved column for mapping a user name to a customization value, such as department.

RPT_JOBMART_RAW

Table 25. RPT_JOBMART_RAW. This table stores LSF job accounting data for reporting.

Column name	Data type	PK	Description
CLUSTER_NAME	VARCHAR(128)	Y	The name of the LSF cluster.
SUBMIT_TIME_GMT	TIMESTAMP		Event expected log time in GMT time zone, presented as the number of seconds after 1970/01/01. This is the submit time of the job.
START_TIME_GMT	TIMESTAMP		Event expected log time in GMT time zone, presented as the number of seconds after 1970/01/01. This is the start time of the job.
FINISH_TIME_GMT	TIMESTAMP		Event expected log time in GMT time zone, presented as the number of seconds after 1970/01/01. This is the finish time of the job.
SUBMIT_TIME	TIMESTAMP		This is the time LSF received the job in submission.
START_TIME	TIMESTAMP		This is the time that the job get started to get executed.
FINISH_TIME	TIMESTAMP	Y	This is the time that the job get finished.
FINISH_ISO_WEEK	VARCHAR(10)		In the format to_char(FINISH_TIME, 'IYYY-IW')

Table 25. RPT_JOBMART_RAW (continued). This table stores LSF job accounting data for reporting.

Column name	Data type	PK	Description
PROJECT_NAME	VARCHAR(4000)		The name of the project.
QUEUE_NAME	VARCHAR(128)		The name of the job queue to which the job was submitted.
USER_GROUP	VARCHAR(512)		The user group of the user who submitted this job.
USER_NAME	VARCHAR(128)		The user name of the user who submitted this job.
JOB_TYPE	VARCHAR(30)		Reserved column. Not in use.
JOB_GROUP	VARCHAR(4000)		The job group under which the job runs.
SLA_TAG	VARCHAR(512)		The SLA service class name under which the job runs.
RES_REQ	VARCHAR(4000)		The resource requirements of this job.
MEM_REQ	NUMBER(10)		The resource requirements of this job.
SUBMISSION_HOST	VARCHAR(512)		The name of the host that submitted this job.
EXEC_HOSTNAME	VARCHAR(512)	Y	The name of the execution host.
EXEC_HOSTTYPE	VARCHAR(128)		The host type of the execution host.
EXEC_HOSTMODEL	VARCHAR(128)		The host model of the execution host.
EXEC_HOSTGROUP	VARCHAR(128)		The group name of the execution host.
NUM_EXEC_PROCS	NUMBER(4)		The number of processors that the job initially requested for execution.
NUMBER_OF_JOBS	NUMBER(19,4)		Number of jobs. In the RPT_JOBMART_RAW, this column is always 1.
NUM_SLOTS	NUMBER(10)		The actual number of slots used for job execution.
JOB_EXIT_STATUS	VARCHAR(32)		The exit status of the job. For further details of these exit status codes, see < lsbatch/lsbatch.h >.
JOB_EXIT_CODE	NUMBER(10)		The exit code of the job.
APPLICATION_NAME	VARCHAR(512)		The application tag assigned to this job.
JOB_ID	NUMBER(15)	Y	The LSF-assigned job ID.
JOB_ARRAY_INDEX	NUMBER(15)	Y	The job array index.
JOB_NAME	VARCHAR(4000)		The name of this job.
JOB_CMD	VARCHAR(10000)		The job command.
JOB_PEND_TIME	NUMBER(19,4)		This is calculated as START_TIME – START_TIME, if START_TIME is not null else it is calculated as FINISH_TIME – START_TIME. The result is in seconds.
JOB_RUN_TIME	NUMBER(19,4)		This is calculated as the time difference between the FINISH_TIME and START_TIME. The result is in seconds.
JOB_TURNAROUND_TIME	NUMBER(19,4)		This is calculated as the time difference between the FINISH_TIME and SUBMIT_TIME. The result is in seconds.
JOB_MEM_USAGE	NUMBER(19,4)		The MEM_USAGE is based on a field from raw table. The subfield is MAX_RMEM. It is in kilobytes.
JOB_SWAP_USAGE	NUMBER(19,4)		The SWAP_USAGE is based on a field from raw table. The subfield is MAX_RSWAP. It is in megabytes.
JOB_CPU_TIME	NUMBER(19,4)		The CPU_TIME is based on two fields from raw table. The subfields are RU_UTIME and RU_STIME. The sum of these two fields is CPU_TIME. It is in seconds.

Table 25. RPT_JOBMART_RAW (continued). This table stores LSF job accounting data for reporting.

Column name	Data type	PK	Description
PEND_TIME	NUMBER(19,4)		The job pending time divided onto execution host, calculated as: `JOB_PEND_TIME * (NUM_SLOTS / NUM_EXEC_PROCS)`
RUN_TIME	NUMBER(19,4)		The job run time divided onto execution host, calculated as: `JOB_RUN_TIME * (NUM_SLOTS / NUM_EXEC_PROCS)`
TURNAROUND_TIME	NUMBER(19,4)		The job turnaround time divided onto execution host, calculated as: `JOB_TURNAROUND_TIME * (NUM_SLOTS / NUM_EXEC_PROCS)`
MEM_USAGE	NUMBER(19,4)		The memory usage of the job on the execution host
SWAP_USAGE	NUMBER(19,4)		The swap usage of the job on the execution host
CPU_TIME	NUMBER(19,4)		The CPU time of the job on the execution host
RANK_MEM	VARCHAR(64)		Rank of job memory usage. For example: '0 GB to 1 GB'
RANK_MEM_REQ	VARCHAR(128)		Rank of job memory requirement. For example: '0 GB to 1 GB'
RANK_RUNTIME	VARCHAR(64)		Rank of job run time. For example: '0 sec to 5 sec'
RANK_PENDTIME	VARCHAR(64)		Rank of job pending time. For example: '0 sec to 5 sec'
RANK_CPUTIME	VARCHAR(64)		Rank of job CPU time.
RANK_EFFICIENCY	VARCHAR(64)		Rank of job efficiency.
JOB_GROUP1	VARCHAR(1024)		The first section of the JOB_GROUP column.
JOB_GROUP2	VARCHAR(1024)		The second section of the JOB_GROUP column.
JOB_GROUP3	VARCHAR(1024)		The third section of the JOB_GROUP column.
JOB_GROUP4	VARCHAR(1024)		The rest of the JOB_GROUP column.
CLUSTER_MAPPING	VARCHAR(4000)		Reserved column for mapping a cluster name to a customization value, such as department.
PROJECT_MAPPING	VARCHAR(4000)		Reserved column for mapping a project name to a customization value, such as department.
USER_MAPPING	VARCHAR(4000)		Reserved column for mapping a user name to a customization value, such as department.
JOB_DESCRIPTION	VARCHAR(4096)		A text description of the job.

RPT_JOBMART_DAY

Table 26. RPT_JOBMART_DAY. This table stores daily LSF job accounting data for reporting. This is grouped by all available dimensions so that the RAW to DAY rollup matches. All values are AVG unless otherwise stated.

Column name	Data type	PK	Description
CLUSTER_NAME	VARCHAR(128)	Y	The name of the LSF cluster.
SUBMIT_TIME_GMT	TIMESTAMP	Y	Event expected log time in GMT timezone, presented as the number of seconds after 1970/01/01. This is the submit time of the job.
START_TIME_GMT	TIMESTAMP	Y	Event expected log time in GMT timezone, presented as the number of seconds after 1970/01/01. This is the start time of the job.
FINISH_TIME_GMT	TIMESTAMP	Y	Event expected log time in GMT timezone, presented as the number of seconds after 1970/01/01. This is the finish time of the job.

Table 26. RPT_JOBMART_DAY (continued). This table stores daily LSF job accounting data for reporting. This is grouped by all available dimensions so that the RAW to DAY rollup matches. All values are AVG unless otherwise stated.

Column name	Data type	PK	Description
SUBMIT_TIME	TIMESTAMP	Y	The time LSF received the job in submission.
START_TIME	TIMESTAMP	Y	The time that the job get started to get executed.
FINISH_TIME	TIMESTAMP	Y	The time that the job get finished.
FINISH_ISO_WEEK	VARCHAR(10)		In the format `to_char(FINISH_TIME, 'IYYY-IW')`.
PROJECT_NAME	VARCHAR(4000)	Y	The name of the project.
QUEUE_NAME	VARCHAR(128)	Y	The name of job queue to which the job was submitted.
USER_GROUP	VARCHAR(512)	Y	The user group of the user who submitted this job.
USER_NAME	VARCHAR(128)	Y	The user name of the user who submitted this job.
JOB_TYPE	VARCHAR(30)	Y	Reserved column. Not in use.
JOB_GROUP	VARCHAR(4000)	Y	The job group under which the job runs.
SLA_TAG	VARCHAR(512)	Y	The SLA service class name under which the job runs.
SUBMISSION_HOST	VARCHAR(512)	Y	The name of the host that submitted this job.
EXEC_HOSTNAME	VARCHAR(512)	Y	The name of the execution host.
EXEC_HOSTTYPE	VARCHAR(128)	Y	The host type of the execution host.
EXEC_HOSTMODEL	VARCHAR(128)	Y	The host model of the execution host.
EXEC_HOSTGROUP	VARCHAR(128)	Y	The group name of the execution host.
NUM_EXEC_PROCS	NUMBER(4)	Y	The number of processors that the job initially requested for execution.
NUMBER_OF_JOBS	NUMBER(19,4)		Number of jobs fall into the group.
NUM_SLOTS	NUMBER(10)		The actual number of slots used for job execution.
JOB_EXIT_STATUS	VARCHAR(32)	Y	The exit status of the job. For further details of these exit status codes, see < lsbatch/lsbatch.h >.
JOB_EXIT_CODE	NUMBER(10)	Y	The exit code of the job.
APPLICATION_NAME	VARCHAR(512)	Y	The application tag assigned to this job.
JOB_PEND_TIME_MAX	NUMBER(19,4)		Calculated from the JOB_PEND_TIME in the RPT_JOBMART_RAW table. It is the maximum JOB_PEND_TIME of the group within the day.
JOB_PEND_TIME_MIN	NUMBER(19,4)		Calculated from the JOB_PEND_TIME in the RPT_JOBMART_RAW table. It is the minimum JOB_PEND_TIME of the group within the day.
JOB_RUN_TIME	NUMBER(19,4)		The total JOB_RUN_TIME of the group within the day.
JOB_RUN_TIME_MAX	NUMBER(19,4)		Calculated from the JOB_RUN_TIME in the RPT_JOBMART_RAW table. It is the maximum JOB_RUN_TIME of the group within the day.
JOB_RUN_TIME_MIN	NUMBER(19,4)		Calculated from the JOB_RUN_TIME in the RPT_JOBMART_RAW table. It is the minimum JOB_RUN_TIME of the group within the day.
JOB_TURNAROUND_TIME_MAX	NUMBER(19,4)		Calculated from the JOB_TURNAROUND_TIME in the RPT_JOBMART_RAW table. It is the maximum JOB_TURNAROUND_TIME of the group within the day.
JOB_TURNAROUND_TIME_MIN	NUMBER(19,4)		Calculated from the JOB_TURNAROUND_TIME in the RPT_JOBMART_RAW table. It is the minimum JOB_TURNAROUND_TIME of the group within the day.

Table 26. RPT_JOBMART_DAY (continued). This table stores daily LSF job accounting data for reporting. This is grouped by all available dimensions so that the RAW to DAY rollup matches. All values are AVG unless otherwise stated.

Column name	Data type	PK	Description
JOB_MEM_USAGE_MAX	NUMBER(19,4)		Calculated from the JOB_MEM_USAGE in the RPT_JOBMART_RAW table. It is the maximum JOB_MEM_USAGE of the group within the day.
JOB_MEM_USAGE_MIN	NUMBER(19,4)		Calculated from the JOB_MEM_USAGE in the RPT_JOBMART_RAW table. It is the minimum JOB_MEM_USAGE of the group within the day.
JOB_SWAP_USAGE_MAX	NUMBER(19,4)		Calculated from the JOB_SWAP_USAGE in the RPT_JOBMART_RAW table. It is the maximum JOB_SWAP_USAGE of the group within the day.
JOB_SWAP_USAGE_MIN	NUMBER(19,4)		Calculated from the JOB_SWAP_USAGE in the RPT_JOBMART_RAW table. It is the minimum JOB_SWAP_USAGE of the group within the day.
JOB_CPU_TIME_MAX	NUMBER(19,4)		Calculated from the JOB_CPU_TIME in the RPT_JOBMART_RAW table. It is the maximum JOB_CPU_TIME of the group within the day.
JOB_CPU_TIME_MIN	NUMBER(19,4)		Calculated from the JOB_CPU_TIME in the RPT_JOBMART_RAW table. It is the minimum JOB_CPU_TIME of the group within the day.
JOB_RUN_EFFICIENCY_MAX	NUMBER(19,4)		Calculated from the JOB_RUN_EFFICIENCY in the RPT_JOBMART_RAW table. It is the maximum JOB_RUN_EFFICIENCY of the group within the day.
JOB_RUN_EFFICIENCY_MIN	NUMBER(19,4)		Calculated from the JOB_RUN_EFFICIENCY in the RPT_JOBMART_RAW table. It is the minimum JOB_RUN_EFFICIENCY of the group within the day.
PEND_TIME	NUMBER(19,4)		Calculated from the PEND_TIME in the RPT_JOBMART_RAW table. It is the total PEND_TIME of the group within the day.
RUN_TIME	NUMBER(19,4)		Calculated from the RUN_TIME in the RPT_JOBMART_RAW table. It is the total RUN_TIME of the group within the day.
TURNAROUND_TIME	NUMBER(19,4)		Calculated from the TURNAROUND_TIME in the RPT_JOBMART_RAW table. It is the total TURNAROUND_TIME of the group within the day.
MEM_USAGE	NUMBER(19,4)		Calculated from the MEM_USAGE in the RPT_JOBMART_RAW table. It is the total MEM_USAGE of the group within the day.
MEM_USAGE_MAX	NUMBER(19,4)		Calculated from the MEM_USAGE in the RPT_JOBMART_RAW table. It is the maximum MEM_USAGE of the group within the day.
MEM_USAGE_MIN	NUMBER(19,4)		Calculated from the MEM_USAGE in the RPT_JOBMART_RAW table. It is the minimum MEM_USAGE of the group within the day.
SWAP_USAGE	NUMBER(19,4)		Calculated from the SWAP_USAGE in the RPT_JOBMART_RAW table. It is the total SWAP_USAGE of the group within the day.
SWAP_USAGE_MAX	NUMBER(19,4)		Calculated from the SWAP_USAGE in the RPT_JOBMART_RAW table. It is the maximum SWAP_USAGE of the group within the day.
SWAP_USAGE_MIN	NUMBER(19,4)		Calculated from the SWAP_USAGE in the RPT_JOBMART_RAW table. It is the minimum SWAP_USAGE of the group within the day.

Table 26. RPT_JOBMART_DAY (continued). This table stores daily LSF job accounting data for reporting. This is grouped by all available dimensions so that the RAW to DAY rollup matches. All values are AVG unless otherwise stated.

Column name	Data type	PK	Description
CPU_TIME	NUMBER(19,4)		Calculated from the CPU_TIME in the RPT_JOBMART_RAW table. It is the total CPU_TIME of the group within the day.
CPU_TIME_MAX	NUMBER(19,4)		Calculated from the CPU_TIME in the RPT_JOBMART_RAW table. It is the maximum CPU_TIME of the group within the day.
CPU_TIME_MIN	NUMBER(19,4)		Calculated from the CPU_TIME in the RPT_JOBMART_RAW table. It is the minimum CPU_TIME of the group within the day.
RUN_EFFICIENCY_MAX	NUMBER(19,4)		Calculated from the RUN_EFFICIENCY in the RPT_JOBMART_RAW table. It is the maximum RUN_EFFICIENCY of the group within the day.
RUN_EFFICIENCY_MIN	NUMBER(19,4)		Calculated from the RUN_EFFICIENCY in the RPT_JOBMART_RAW table. It is the minimum RUN_EFFICIENCY of the group within the day.
RANK_MEM	VARCHAR(64)	Y	Rank of job memory usage. For example: '0 GB to 1 GB'
RANK_RUNTIME	VARCHAR(64)	Y	Rank of job pending time. For example: '0 sec to 5 sec'
RANK_PENDTIME	VARCHAR(64)	Y	Rank of job pending time. For example: '0 sec to 5 sec'
RANK_CPUTIME	VARCHAR(64)	Y	Rank of job CPU time.
RANK_EFFICIENCY	VARCHAR(64)	Y	Rank of job efficiency.
CLUSTER_MAPPING	VARCHAR(4000)		Reserved column for mapping a cluster name to a customization value, such as department.
PROJECT_MAPPING	VARCHAR(4000)		Reserved column for mapping a project name to a customization value, such as department.
USER_MAPPING	VARCHAR(4000)		Reserved column for mapping a user name to a customization value, such as department.
JOB_DESCRIPTION	VARCHAR(4096)		A text description of the job.

RPT_WORKLOAD_STATISTICS_RAW

Table 27. RPT_WORKLOAD_STATISTICS_RAW. This table stores daily LSF job statistics data for reporting.

Column name	Data type	PK	Description
TIME_STAMP	TIMESTAMP	Y	Sampling time in the local cluster time zone.
ISO_WEEK	VARCHAR(10)		In the format `TO_CHAR(TIME_STAMP, 'IYYY-IW')`.
CLUSTER_NAME	VARCHAR(128)	Y	The name of the LSF cluster.
TIME_STAMP_GMT	TIMESTAMP		Sampling time in GMT time zone, presented as the number of seconds after 1970/01/01.
JOB_STATUS_STR	VARCHAR(256)	Y	The LSF job status string, such as `PEND`, `RUN`.
HOST_NAME	VARCHAR(512)	Y	The name of the execution host in the cluster.
HOST_TYPE	VARCHAR(128)		The type the execution host, such as UltraSparc10.
HOST_MODEL	VARCHAR(128)		The model the execution host, such as Linux86.
HOST_GROUP	VARCHAR(128)		The LSF host group to which the execution host belongs.
PROJECT_NAME	VARCHAR(4000)	Y	The project name under which the job runs; otherwise, it is `'-'`.
QUEUE_NAME	VARCHAR(128)	Y	The queue name under which the job runs; otherwise, it is `'-'`.

Table 27. RPT_WORKLOAD_STATISTICS_RAW (continued). This table stores daily LSF job statistics data for reporting.

Column name	Data type	PK	Description
USER_GROUP	VARCHAR(512)	Y	The user group under which the job runs; otherwise, it is '-'.
USER_NAME	VARCHAR(128)	Y	The user who submitted the job.
JOB_GROUP	VARCHAR(4000)	Y	The job group under which the job runs; otherwise, it is '-'.
APPLICATION_NAME	VARCHAR(512)	Y	The application tag assigned to this job
NUM_PROCESSORS	NUMBER(15)	Y	The number of slots required to run the job.
NUMBER_SLOTS	NUMBER(19,4)		The total number of slots used by jobs in this time sample with the same PK.
NUMBER_JOBS	NUMBER(19,4)		The total number of jobs in this time sample with the same PK.
NUMBER_JOB_HOSTS	NUMBER(19)		The total number of execution hosts for each job in this time sample.
SWAP_USAGE	NUMBER(19,4)		The sum of maximum swap used by jobs in this time sample with the same PK.
MEM_USAGE	NUMBER(19,4)		The sum of maximum memory used by jobs in this time sample with the same PK.
CPU_DELTA	NUMBER(19,4)		The sum of CPU delta for all jobs in this sample with the same PK.
STATUS_DURATION	NUMBER(19,4)		How long the job has remained in the current status in the sample. For a job run on multiple hosts, it is split by the calculation: (slots used on the host) / (total slots used by the job)
INTERVAL_PERIOD	NUMBER(19,4)		The sampling interval in seconds.
CLUSTER_MAPPING	VARCHAR(4000)		Reserved column for mapping a cluster name to a customization value, like department
PROJECT_MAPPING	VARCHAR(4000)		Reserved column for mapping a project name to a customization value, like department
USER_MAPPING	VARCHAR(4000)		Reserved column for mapping a user name to a customization value, like department
JOB_DESCRIPTION	VARCHAR(4096)		The text description of the job.

RPT_JOB_PENDINGREASON_RAW

Table 28. RPT_JOB_PENDINGREASON_RAW. This table stores data about pending job instances for reporting.

Column name	Data type	PK	Description
TIME_STAMP_GMT	TIME_STAMP_GMT		Sampling time in GMT timezone, presented as the number of seconds after 1970/01/01.
TIME_STAMP	TIMESTAMP	Y	Sampling time in the local cluster timezone.
ISO_WEEK	VARCHAR(10)		In the format TO_CHAR(TIME_STAMP, 'IYYY-IW').
CLUSTER_NAME	VARCHAR(128)	Y	The name of the LSF cluster.
USER_NAME	VARCHAR(128)	Y	The user who submitted the job.
PROJECT_NAME	VARCHAR(4000)	Y	The project name that the job belongs to.
QUEUE_NAME	VARCHAR(128)	Y	The queue name that the job belongs to.
APPLICATION_NAME	VARCHAR(512)	Y	The application tag assigned to this job; otherwise, '-'.
HOST_TYPE	VARCHAR(128)	Y	Type of the job submission host.
PENDING_REASON	VARCHAR(4000)	Y	The reason why the job is in the PEND or PSUSP state.

Table 28. RPT_JOB_PENDINGREASON_RAW (continued). This table stores data about pending job instances for reporting.

Column name	Data type	PK	Description
PENDING_REASON_TYPE	VARCHAR(4000)	Y	The pending reason type, such as: Job Related Reasons Queue and System Related Reasons User Related Reasons Host Related Reasons MC Related Reasons Other Reasons
PENDING_TIME_RANK	VARCHAR(4000)	Y	Rank of job pending time. For example: '0 sec to 5 sec'.
CLUSTER_MAPPING	VARCHAR(4000)		Reserved column for mapping a cluster name to a customization value, such as department.
PROJECT_MAPPING	VARCHAR(4000)		Reserved column for mapping a project name to a customization value, such as department.
USER_MAPPING	VARCHAR(4000)		Reserved column for mapping a user name to a customization value, such as department.
NUM_JOBS	NUMBER(15)		The total number of jobs in the group at the sampling point.

RPT_FLEXLM_LICUSAGE_RAW

Table 29. RPT_FLEXLM_LICUSAGE_RAW. This table stores data about FlexLM license usage for reporting.

Column name	Data type	PK	Description
TIME_STAMP	VARCHAR(128)	Y	The record sample time in the local cluster time zone.
TIME_STAMP_GMT	NUMBER(13)	Y	Event expected log time in GMT timezone.
ISO_WEEK	VARCHAR (10)		In the format `TO_CHAR(TIME_STAMP, 'IYYY-IW')`.
LIC_SITE_NAME	VARCHAR(256)	Y	The user-specified server name.
LIC_SERVER_MASTER	VARCHAR(128)		The number of license requested.
LIC_SERVER_NAME	VARCHAR(128)	Y	The user-specified server name. If not specified, it is stored as `'-'`.
LIC_VENDOR_NAME	VARCHAR(128)	Y	The license vendor.
LIC_FEATURE_NAME	VARCHAR(128)	Y	The license feature name.
LIC_VERSION	VARCHAR(128)	Y	The license version. If not specified, it is stored as `'-'`.
USER_NAME	VARCHAR(128)	Y	The user who tried to check out the license.
HOST_NAME	VARCHAR(256)	Y	The host the user is logged onto.
LIC_TOTAL	NUMERIC(19,4)		The total usage of this license.
LIC_USAGE	NUMERIC(19,4)		The usage of this license. If there is no usage, it will be 0.
LIC_RESERVATION	NUMERIC(19,4)		The reservation of this license. If there is no reservation, it will be 0.
LIC_CONSUMPTION	NUMERIC(19,4)		Calculated as LIC_USAGE × used minutes.
FACTOR_BY_SERVER	NUMERIC(19,4)		Reserved column for calculation of total license by server.
TOTAL_BY_FEATURE	NUMERIC(19,4)		Reserved column for calculation of total license by feature. The calculation is based on all of the current checkouts which, when summed, equates to the number of licenses.
INTERVAL_PERIOD	NUMBER(15)		The sampling interval, in seconds.
USER_MAPPING	VARCHAR(4000)		Reserved column for mapping a user name to a customization value, such as department.

RPT_FNM_LICUSAGE_RAW

Table 30. RPT_FNM_LICUSAGE_RAW. This table stores "hourly job & license feature" level FlexNet Manager license usage information for all the LSF and non-LSF jobs.

Column name	Data type	PK	Description
TIME_STAMP	TIMESTAMP NOT NULL ENCODING COMMONDELTA_COMP	PK	The simulated hourly level sampling time in the **FNMTimeZone** time zone (as configured in `fnmloader.properties`), or, if **FNMTimeZone** is null, in the local cluster time zone.
TIME_STAMP_GMT	TIMESTAMP NOT NULL ENCODING COMMONDELTA_COMP		The simulated hourly level sampling time in GMT time.
ISO_WEEK	VARCHAR(10) ENCODING RLE		In the format `TO_CHAR(TIME_STAMP, 'IYYY-IW')`.
CLUSTER_NAME	VARCHAR(128) ENCODING RLE		Set to `'-'` for non-LSF license usage data.
CLUSTER_MAPPING	VARCHAR(4000) ENCODING RLE		Reserved column for mapping a cluster name to a customization value, such as department.
JOB_ID	NUMBER(15) NOT NULL ENCODING DELTAVAL		Set to `-1` for non-LSF license usage data.
JOB_ARRAY_INDEX	NUMBER(15) NOT NULL ENCODING RLE		Set to `-1` for non-LSF license usage data.
LIC_SERVER_NAME	VARCHAR(128) NOT NULL ENCODING RLE	PK	Handle redundant license servers at loader side. For example, to keep consistent for `s1:s2:s3`, `s2:s1:s3`, and so on.
LIC_VENDOR_NAME	VARCHAR(128) NOT NULL ENCODING RLE	PK	The name of the license vendor.
LIC_FEATURE_NAME	VARCHAR(128) NOT NULL ENCODING RLE	PK	The name of the license feature.
LIC_SITE_NAME	VARCHAR(256) NOT NULL ENCODING RLE	PK	The user-specified server name.
LIC_VERSION	VARCHAR(128) NOT NULL ENCODING RLE	PK	The license version. If not specified, it is stored as `'-'`.
LIC_PROJECT	VARCHAR(60) NOT NULL ENCODING RLE	PK	Limited to 30 characters at license server side.
SUBMIT_TIME	TIMESTAMP ENCODING COMMONDELTA_COMP	PK	Set to `'1970-1-1'` for non-LSF license us age data.
USER_NAME	VARCHAR(128) ENCODING RLE		Name of the user that used the license.
USER_MAPPING	VARCHAR(4000) ENCODING RLE		Reserved column for mapping a user name to a customization value, such as department.
PROJECT_NAME	VARCHAR(4000) ENCODING RLE		The name of the project.
PROJECT_MAPPING	VARCHAR(4000) ENCODING RLE		Reserved column for mapping a project name to a customization value, such as department.
HOST_NAME	VARCHAR(128) ENCODING RLE		Name of the host on which the license was used.

Table 30. RPT_FNM_LICUSAGE_RAW (continued). This table stores "hourly job & license feature" level FlexNet Manager license usage information for all the LSF and non-LSF jobs.

Column name	Data type	PK	Description
MIN_LIC_USAGE	NUMBER(19,4)		Minimum number of used licenses within the hour.
MAX_LIC_USAGE	NUMBER(19,4)		Maximum number of used licenses within the hour.
AVG_LIC_USAGE	NUMBER(19,4)		Average number of used licenses weighted by usage duration, calculated as: `Sum(Lic_Usage × (Checkin - Checkout)) / 1 hour`
USED_MINUTES	NUMBER(19,4)		Total minutes of license consumption within the hour, calculated as: `Sum(Lic_usage × (Checkin - Checkout))`
LIC_TOTAL	NUMBER(15)		Total number of licenses got from flexnet_license_info table Max(lic_num) of the license server/vendor/feature in the simulated sampling hour.
FACTOR_BY_SERVER	NUMBER(19,4)		Total licenses by server divided by count of sampling instances of the license server in the hour, calculated as: `(1.0 * LIC_TOTAL) / COUNT(*) OVER(PARTITION BY TIME_STAMP, LIC_SERVER_NAME, LIC_VENDOR_NAME, LIC_FEATURE_NAME)`
TOTAL_BY_FEATURE	NUMBER(19,4)		Total licenses of the feature across different license servers, calculated as: `SUM(FACTOR_BY_SERVER) OVER(PARTITION BY TIME_STAMP, LIC_VENDOR_NAME, LIC_FEATURE_NAME)`

RPT_FNM_LICUSAGE_BY_FEATURE

Table 31. RPT_FNM_LICUSAGE_BY_FEATURE. This table stores "hourly & license feature" level average and peak FlexNet Manager license usage information.

Column name	Data type	Key (PK/ FK)	Description
TIME_STAMP	TIMESTAMP NOT NULL ENCODING COMMONDELTA_COMP	PK	The simulated hourly level sampling time in the **FNMTimeZone** time zone (as configured in `fnmloader.properties`), or, if **FNMTimeZone** is null, in the local cluster time zone.
TIME_STAMP_GMT	TIMESTAMP NOT NULL ENCODING COMMONDELTA_COMP		The simulated hourly level sampling time in GMT time.
ISO_WEEK	VARCHAR(10)		In the format `TO_CHAR(TIME_STAMP, 'IYYY-IW')`.
LIC_VENDOR_NAME	VARCHAR(128) NOT NULL ENCODING RLE	PK	The name of the license vendor.
LIC_FEATURE_NAME	VARCHAR(128) NOT NULL ENCODING RLE	PK	The name of the license feature.
AVG_ USAGE	NUMBER(19,4)		Average number of used licenses weighted by usage duration, calculated as: `Sum(Lic_Usage × (Checkin - Checkout)) / 1 hour`

Table 31. RPT_FNM_LICUSAGE_BY_FEATURE (continued). This table stores "hourly & license feature" level average and peak FlexNet Manager license usage information.

Column name	Data type	Key (PK/ FK)	Description
PEAK_ USAGE	NUMBER(19,4)		Maximum number of used license within the hour.
TOTAL_BY_FEATURE	NUMBER(19,4)		Total licenses for the feature across different license servers.

RPT_FNM_LICUSAGE_BY_SERVER

Table 32. RPT_FNM_LICUSAGE_BY_SERVER. This table stores "hourly & license feature & license server" level average and peak license usage information.

Column name	Data type	Key (PK/ FK)	Description
TIME_STAMP	TIMESTAMP NOT NULL ENCODING COMMONDELTA_COMP	PK	The simulated hourly level sampling time in the **FNMTimeZone** time zone (as configured in `fnmloader.properties`), or, if **FNMTimeZone** is null, in the local cluster time zone.
TIME_STAMP_GMT	TIMESTAMP NOT NULL ENCODING COMMONDELTA_COMP		The simulated hourly level sampling time in GMT time
ISO_WEEK	VARCHAR(10)		In the format `TO_CHAR(TIME_STAMP, 'IYYY-IW')`.
LIC_SERVER_NAME	VARCHAR(128) NOT NULL ENCODING RLE	PK	Handle redundant license servers at loader side. For example, to keep consistent for `s1:s2:s3`, `s2:s1:s3`, and so on.
LIC_VENDOR_NAME	VARCHAR(128) NOT NULL ENCODING RLE	PK	The name of the license vendor.
LIC_FEATURE_NAME	VARCHAR(128) NOT NULL ENCODING RLE	PK	The name of the license feature.
TOTAL_BY_SERVER	NUMBER(19,4)		Total licenses for the feature across different license servers.
TOTAL_BY_FEATURE	NUMBER(19,4)		Total licenses by feature across different license servers.

RPT_LICENSE_DENIALS_RAW

Table 33. RPT_LICENSE_DENIALS_RAW. This table stores information about license denials for reporting.

Column name	Data type	Key (PK/ FK)	Description
TIME_STAMP	TIMESTAMP	PK	Event expected log time in the local cluster time zone.
FNM	NUMBER(1)		Reports FlexNet Manager (FNM) denial events and license denials.
TIME_STAMP_GMT	TIMESTAMP	PK	Event expected log time in GMT time zone.
ISO_WEEK	VARCHAR(10)		In the format `TO_CHAR(TIME_STAMP, 'IYYY-IW')`.
PLC_ID	VARCHAR(20)		The ID of the data loader controller. Each controller has a unique ID among all the clusters.
USER_NAME	VARCHAR(128)	PK	The name of the user who requested the license.
HOST_NAME	VARCHAR(128)	PK	The name of the host the user is logged onto.

Table 33. RPT_LICENSE_DENIALS_RAW (continued). This table stores information about license denials for reporting.

Column name	Data type	Key (PK/ FK)	Description
LIC_SERVER_NAME	VARCHAR(128)	PK	Handle redundant license servers at loader side. For example, to keep consistent for `s1:s2:s3`, `s2:s1:s3`, and so on.
LIC_VENDOR_NAME	VARCHAR(128)	PK	The name of the license vendor.
LIC_FEATURE_NAME	VARCHAR(128)	PK	The name of the license feature.
LIC_SITE_NAME	VARCHAR(256)	PK	The user-specified server name.
DENIALS	NUMBER(15)		Total number of license denials.
LIC_PROJECT	VARCHAR(60)		In Platform Analytics version 8.0.2 and earlier, the maximum length of this column is 60; it has been enlarged to 255 for consistence with FlexNet Manager.
SUBMISSION_TIME	NUMBER(15)		The time the job was submitted.
JOB_ID	NUMBER(15)		The LSF-assigned job ID.
JOB_ARRAY_ID	NUMBER(15)		The ID assigned to the LSF job array.
CLUSTER_NAME	VARCHAR(128)		The name of the cluster.
LIC_VERSION	VARCHAR(128)		The license version. If not specified, it is stored as `'-'`.
PROJECT_NAME	VARCHAR(128)		The project name of the job array, from the `-P` option in bsub.

Appendix B. Business data mapping

Platform Analytics provides a way to add business data mappings to report data for clusters, projects, and users.

Business data is the business relationship between the cluster, project, and user dimensions to information that is external to Platform Analytics data collection. For example, the data collected by Platform Analytics would show that a user ran a workload, but would not show the department to which that user belongs, as this relationship is external business data. You can use a business data mapping to map user names to departments and have this information available for reporting.

Platform Analytics stores mapping data for clusters, projects, and users in three standard tables, respectively, as shown in Table 34.

Table 34. Business data mapping tables

Report dimension	Mapping table
Cluster	SYS_CLUSTERNAME_MAPPING (CLUSTER_NAME, CLUSTER_MAPPING)
Project	SYS_PROJECTNAME_MAPPING (PROJECT_NAME, PROJECT_MAPPING)
User	SYS_USERNAME_MAPPING (USER_NAME, USER_MAPPING)

The mappings that you specify in the mapping tables are based on your business needs. For instance, you can choose to map users to departments, projects to divisions, and clusters to regions.

There are two types of business data mapping, *static* and *dynamic*.

Static business data mapping

Static business data mapping applies the mappings during data transformation. When the raw data is transformed into report data, Platform Analytics queries the cluster, project, and user mapping tables and adds the mapped data into the report data tables.

No special customization is needed for static business data mapping, other than maintaining the information in the mapping tables.

Static business data mappings remain as they were at the time the report data was transformed. Therefore, changing or adding to the mappings in the mapping tables does not affect the mappings in historical report data. This is often the preferred type of business data mapping.

Example: If user *User1* was in department *DepartmentA*, and then moves to *DepartmentB*, any workload that *User1* ran while in *DepartmentA* will always be historically reported as being for *DepartmentA*. When the user mapping table is updated to show that *User1* is now in *DepartmentB*, then, from that point forward, any workload run by *User1* will be reported as being for *DepartmentB*.

As mappings are added or changed over time, the only way to update the mappings of historical data to reflect the current mappings is to re-aggregate all of the data for that ETL flow.

Implementing static business data mapping

Perform this task to set up and use static business data mapping.

About this task

This procedure uses static user data mapping to illustrate how to implement static business data mapping. The procedure works similarly for cluster and project data mapping.

Procedure

1. Add the mapping data into the mapping table.

   For user data mapping, the standard mapping table is SYS_USERNAME_MAPPING. This example maps users to departments.

Table 35. Example of the SYS_USERNAME_MAPPING table

USER_NAME	USER_MAPPING
User1	DepartmentA
User2	DepartmentA
User3	DepartmentB

2. When the ETLs run to transform the raw data into report data, the user data mappings will also be added into the report tables.

   Figure 10 illustrates the user data mapping during the Workload Accounting data transformation process.

Figure 10. Example of user data transformation and static mapping

3. In the workbook **Dimensions** pane, rename the **USER_MAPPING** field to an appropriate name, such as **Department**.

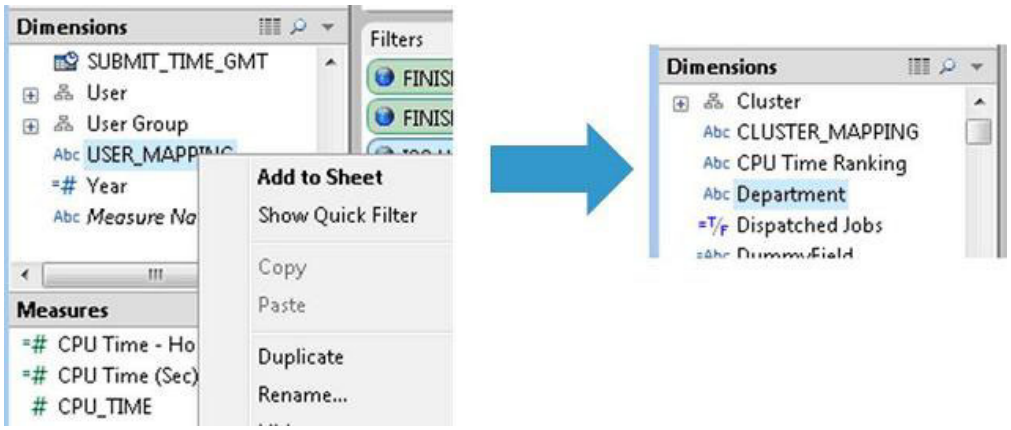

Figure 11. Example of renaming the **USER_MAPPING** *field to* **Department** *in the Workload Accounting workbook*

Results

You can now use the new **Department** business data mapping in worksheets within the workbook.

Dynamic business data mapping

With dynamic business data mapping, the mapping occurs at the report level, rather than at the data transformation level. The data is mapped dynamically by doing a table join on the report data table and the mapping table in the workbook. The mappings are not maintained historically in the report data table.

Dynamic business data mapping allows you to always report on the mappings that are defined at the time you run the report. By changing the mappings in the mapping tables, you can get different views of the same report. For example, by changing the project mapping, you can restructure how projects are grouped and see this view for all historical data.

For environments with small to medium job throughput, it is possible to implement dynamic business data mapping simply by joining the tables in the workbook without any additional customization and without much of an impact on performance.

Note: This approach is not suitable for big data deployments.

For environments with large job throughput, additional customization is necessary to maintain performance. This involves creating foreign keys on the mapping tables and custom joined table projections.

The advanced process is described in the following section.

Implementing dynamic business data mapping

Perform these tasks to set up and use dynamic business data mapping.

About this task

Consider the example of a Workload Accounting Daily workbook to understand how dynamic business data mapping is done at the report level. The general steps to implement dynamic mapping are:

1. Create or maintain a mapping table, such as SYS_PROJECTNAME_MAPPING.
2. Associate the RPT_JOBMART_DAY report table with the SYS_PROJECTNAME_MAPPING mapping table.
3. Generate a new view that contains the mapping relationship.

The specific implementation tasks will use this example.

Modifying the database schema

Perform this task to modify the database schema

Procedure

1. Create the mapping table, such as SYS_PROJECTNAME_MAPPING, if one does not exist.

```
CREATE TABLE SYS_PROJECTNAME_MAPPING
(
PROJECT_NAME VARCHAR(4000) NOT NULL ENCODING RLE,
PROJECT_MAPPING VARCHAR(4000) NOT NULL ENCODING RLE,
PRIMARY KEY(PROJECT_NAME)
)
ORDER BY PROJECT_NAME
SEGMENTED BY HASH(PROJECT_NAME) ALL NODES
KSAFE :K_SAFE
```

   where:

   K_SAFE
   The K-Safe level of the Vertica database is determined by the number of database nodes. Check the actual K-Safe level of your Vertica database and set this value accordingly.

   Note: Data in the mapping tables can be updated but cannot be deleted.

2. Add a foreign key on the related RPT table.

   Add an extra foreign key to create a connection between the target data table and the mapping table. This enhances workbook performance.

```
ALTER TABLE rpt_jobmart_day
ADD CONSTRAINT fk_project FOREIGN KEY
(project_name) REFERENCES
SYS_PROJECTNAME_MAPPING(project_name);
```

Dynamically adding new data to the mapping table

Perform this task to enhance the original ETL by adding a filter for initializing the newly added mapping table.

About this task

The purpose of this task is to automatically add new entries to the mapping table when new project names are found in the reporting table. To accomplish this, you will add a filter to the ETL flow that will:

- Scan the target data table and find all foreign key fields that are in the target data table but that have no value in the mapping table.
- Set a default value for all found item.

This task will use the JobMart Daily flow as an example.

Procedure

1. Edit the main_jobmart_daily.xml file and add the filter, as shown.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE etl SYSTEM "etl.dtd">
<etl Name="WorkloadAccountingDaily" Description="ETL for Workload Accounting report"
Type="Raw">
   <Extractor Class="DependentExtractor" Path="JobmartDailyDepParams.xml"/>
   <Transform Class="Filter" Name="Mapping Filter" Path="mapping_filter.xml" />
   <Loader Class="RecordInsertLoader" Path="RptJobmartDailyLoader.xml" />
</etl>
```

2. Create the mapping_filter.xml file and save it in the same location as the main_jobmart_daily.xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Transform SYSTEM "filter.dtd">
<Transform>
   <Filter Remove="N">
      <Criteria FieldName="CLUSTER_NAME" FieldValue="*"/>
      <etl Path="mapping_filter_etl.xml" />
   </Filter>
</Transform>
```

3. Create the mapping_filter_etl.xml file and save it in the same location as the main_jobmart_daily.xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE etl SYSTEM "etl.dtd">
<etl Name="Mapping etl" Description="Does nothing">
   <Loader Class="RecordInsertLoader" Path="mappingFilterLoader.xml" />
</etl>
```

4. Create the mappingFilterLoader.xml file and save it in the same location as the main_jobmart_daily.xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Loader SYSTEM "jdbcloader.dtd">
<Loader>
   <DataSource MaxTransSize="1" Connection="DEFAULT">ReportDB</DataSource>
      <SQL Type="Insert" Path="../../../../work/Platform_Analytics.Mapping_Filter_ETL.bad">
         <Statement>
            <![CDATA[
               insert into SYS_PROJECTNAME_MAPPING(project_name, project_mapping)
               select t.project_name, 'No mapping' as project_mapping
                from RPT_JOBMART_RAW t left outer join SYS_PROJECTNAME_MAPPING m
                on (t.project_name = m.project_name)
               where m.project_mapping is null and t.cluster_name = ? and
                t.FINISH_TIME >= ? and t.FINISH_TIME < ?
               group by t.project_name;
            ]]>
         </Statement>
         <Parameter FieldName="C1"/>
         <Parameter FieldName="C2"/>
         <Parameter FieldName="C3"/>
      </SQL>
      <Field Name="CLUSTER_NAME" Column="C1"/>
      <Field Name="START_TIME" Column="C2"/>
      <Field Name="END_TIME" Column="C3"/>
</Loader>
```

Modifying a report

Perform this task to modify a report to select multiple tables, instead of a single table, as the data source.

Procedure

1. Edit the report, select the **Multiple Tables** option, and select the RPT_JOBMART_DAY table, as shown in Figure 12 on page 86.

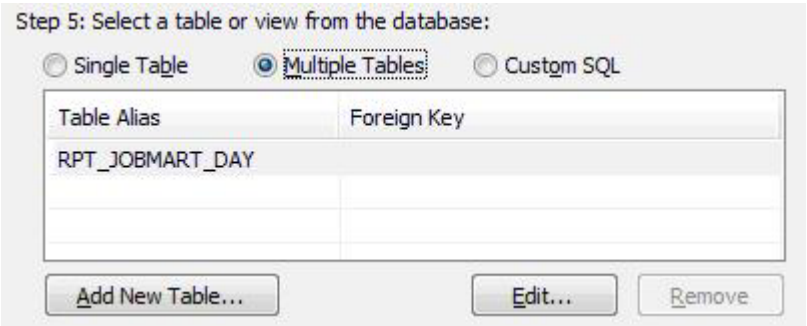

Figure 12. Example of selecting multiple tables for reporting

2. Click **Add New Table...**
3. On the **Table** tab of the **Add Table** dialog, select the `SYS_PROJECTNAME_MAPPING` table, as shown in Figure 13.

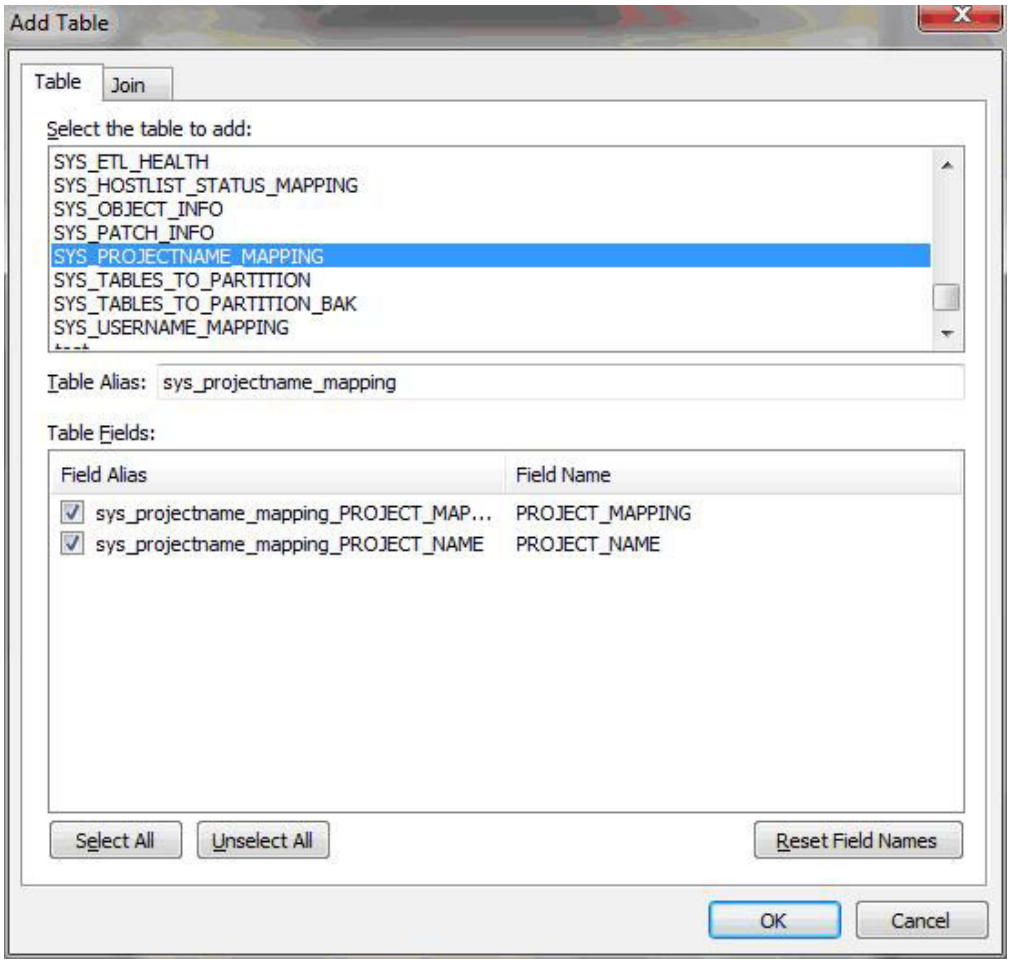

Figure 13. Example of the Add Table dialog: Selecting the table to add

4. On the **Join** tab, edit the **Join Clause**, as shown in Figure 14 on page 87, to be:
   `[RPT_JOBMART_DAY].[PROJECT_NAME] = [sys_projectname_mapping].[PROJECT_NAME]`

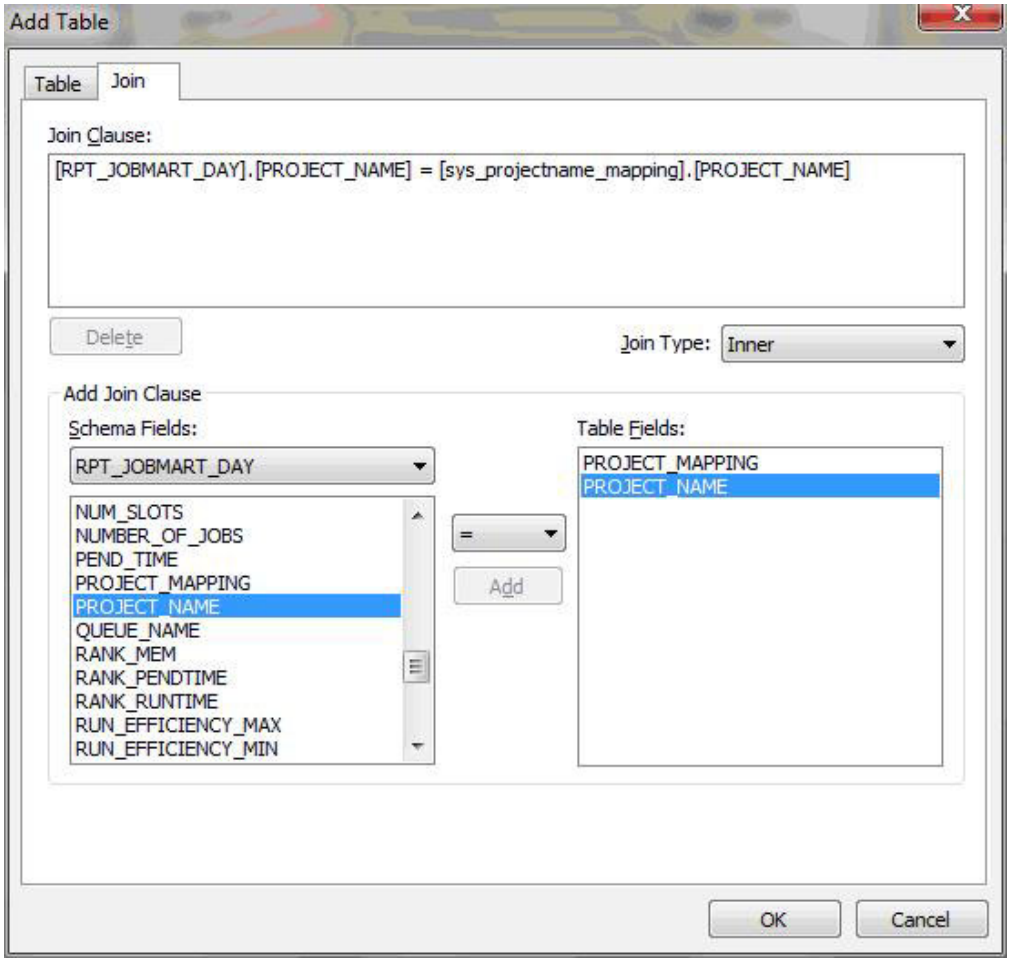

Figure 14. Example of the Add Table dialog: Specifying a Join Clause

5. Click **OK**.

Performance tuning

Perform this task to create a new projection for the target data table to improve performance.

About this task

Since the workbook now uses multiple tables as the data source, creating a new projection will pre-build the connection of the target data table and the mapping table.

The following example shows the general format to create the projection:

```
CREATE PROJECTION PROJECTION_NAME (
   T1_C1,
   T1_C2,
   T1_C3,
   T2_C1
)
AS SELECT T1_C1,
          T1_C2,
          T1_C3,
```

```
          T2_C1
   FROM T1, T2
   WHERE T1.C=T2.C
KSAFE :K_SAFE
```

where:

T1
: Table 1, the target data table

T2
: Table 2, the mapping table

T1_C1, ..., *T2_C1*
: The columns that are shown in the workbook

T1_Cn
: A column from table T1 (the target data table)

T2_Cn
: A column from table T2 (the mapping table)

This task continues to use the JobMart Daily report as the example.

Procedure

Create the projection on the target data table (RPT_JOBMART_DAY) and the project mapping table (SYS_PROJECTNAME_MAPPING), as shown:

```
CREATE PROJECTION PROJ_RPT_JOBMART_DAY_01 (
   FINISH_TIME ENCODING COMMONDELTA_COMP,
   FINISH_ISO_WEEK ENCODING RLE,
   CLUSTER_NAME ENCODING RLE,
   JOB_STATUS_STR ENCODING RLE,
   RANK_MEM ENCODING RLE,
   RANK_RUNTIME ENCODING RLE,
   RANK_PENDTIME ENCODING RLE,
   project_mapping ENCODING RLE,
   NUMBER_OF_JOBS
)
AS SELECT a.FINISH_TIME,
          a.FINISH_ISO_WEEK,
          a.CLUSTER_NAME,
          a.JOB_STATUS_STR,
          a.RANK_MEM,
          a.RANK_RUNTIME,
          a.RANK_PENDTIME,
          b.project_mapping,
          a.NUMBER_OF_JOBS
   FROM RPT_JOBMART_DAY a, SYS_PROJECTNAME_MAPPING b
   WHERE a.PROJECT_NAME=b.project_name
   ORDER BY a.CLUSTER_NAME,
         a.JOB_STATUS_STR,
         a.RANK_MEM,
         a.RANK_RUNTIME,
         a.RANK_PENDTIME,
         b.project_mapping,
         a.FINISH_ISO_WEEK,
         a.FINISH_TIME
      SEGMENTED BY HASH(FINISH_TIME, CLUSTER_NAME) ALL NODES
   KSAFE :K_SAFE
```

Results

The new PROJ_RPT_JOBMART_DAY_01 projection is added on the RPT_JOBMART_DAY and SYS_PROJECTNAME_MAPPING tables.

Maintaining the mapping tables

Performance might suffer when updating the mapping table if the projection based on the mapping table is large. In such cases, use this general procedure to update the mapping table data.

Procedure

1. Drop all projections based on the mapping table.
2. Update the mapping table.
3. Recreate the projections.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web

sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Intellectual Property Law
Mail Station P300
2455 South Road,
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application

programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

LSF, Platform, and Platform Computing are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, or service names may be trademarks or service marks of others.

Printed in USA

SC14-7572-00