
Upgrading Platform Analytics

Platform Analytics
Version 7.6
April 2010



Contents

1	Analytics upgrade packages	3
	About the Analytics upgrade packages	4
2	Prepare for the Analytics upgrade	7
	What you need to do	8
3	Upgrade Analytics	13
	What you need to do	14
4	Optimize the Analytics node	23
	What you need to do	24
5	Optimize the Analytics server	29
	What you need to do	30

Analytics upgrade packages

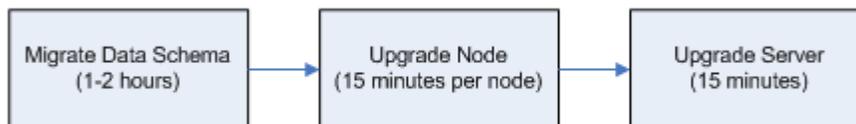
About the Analytics upgrade packages

The Analytics 7.6 upgrade packages allow you to upgrade your Analytics system from version 7.5 to version 7.6. The upgrade packages include updates to the Oracle database schema, the Analytics node, and the Analytics server. You can upgrade Analytics 7.5 with any patches or solution packages installed.

You do not need a separate license for Analytics 7.6 if you already have a license for Analytics 7.5.

The Analytics upgrade packages will help improve the upgrade process:

- The time to migrate the Oracle database schema and Analytics data depends on the performance of your database host and the amount of Analytics-related data in the database. For example, it will take approximately one to two hours to migrate the database schema and data for a cluster with 1.5 million jobs a day, 2000 records per sampling interval, and a data retention period of two weeks for raw data, 26 weeks for hourly data, and 52 weeks for daily data.
- Upgrading the Analytics node will typically take 15 minutes. The data loaders will be down for up to five minutes during this time while being upgraded.
- Upgrading the Analytics server will typically take 15 minutes.



Customization issues

The Analytics 7.6 upgrade packages can handle certain customizations that you applied to certain built-in files.

Refer to *README for Upgrading Platform Analytics (UNIX)* and *README for Upgrading Platform Analytics (Windows)* to see the lists of files in the UNIX or Windows hosts that are changed as a result of the upgrade. If you have customizations in any of the files that will be changed in the Analytics upgrade, you need to make note of these files and check them after the upgrade to see if the customizations are still in place. If they are not, you need to restore the customizations to these files.

Database schema

The following customizations to the database schema will remain in the upgraded database schema:

- New tables added to the Analytics database
- New columns added to built-in tables in the Analytics database

The Analytics database schema upgrade package cannot handle certain customizations to the database schema. After the upgrade, your Analytics node and server hosts may encounter problems, which are sent as events. The following customizations may result in problems after the upgrade:

- Dropped columns in built-in tables
- Modified built-in table names
- Modified column names or types in built-in tables

Analytics node

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

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

Analytics server

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

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

Analytics upgrade packages

Prepare for the Analytics upgrade

Obtain the upgrade package files and upgrade the Oracle database to prepare for the Analytics upgrade.

What you need to do

Check the following:

- The Oracle database host is properly configured and running.
- You have a user name, password, and URL to access the database.

The Analytics database user must have the following:

- Roles: Connect and resource.
- System privileges: Create view.
- The Oracle database and the database host meet the Analytics system requirements.

Refer to the *Release Notes for Platform Analytics* for the latest system requirements for the database host.

1. [Obtain the necessary upgrade package files](#) on page 8
2. [Migrate the database schema](#) on page 8
3. [Optimize the Oracle database](#) on page 9
4. [Verify the bjobs data loader deployment](#) on page 10

Obtain the necessary upgrade package files

1. Obtain the necessary files for upgrading Analytics.
 - Analytics server upgrade package
 - UNIX: `analyticcs7.6_server_upgrade_platform.bin`
 - Windows: `analyticcs7.6_server_upgrade_platform.exe`
 - Analytics node upgrade package
 - UNIX: `analyticcs7.6_node_upgrade_platform.bin`
 - Windows: `analyticcs7.6_node_upgrade_platform.exe`
 - Analytics database schema upgrade package
`analyticcs7.6_upgrade_dbschema.tar`

Migrate the database schema

Migrate the Oracle database schema and the data to prepare the database host for the Analytics upgrade. The typical data migration may take one to two hours, depending on the performance of your database host and the amount of Analytics-related data in the database.

Note:

The database schema upgrade package does not support upgrading non-partitioned schema to a partitioned schema, and vice versa.

Therefore, if you are using a non-partitioned schema, you cannot upgrade to a partitioned schema using the database schema upgrade package.

1. Grant additional, required privileges to the Analytics database user.
 - a) Log into any host with access to the database host as the Analytics database user.

- b) Extract the database schema upgrade package (`analytics7.6_upgrade_dbschema.tar`).
- c) Navigate to the directory containing your database schema scripts.
- d) Use `sqlplus` to run the `grant_privilege.sql` script with DBA privileges to grant additional, required privileges to the Analytics database user.

```
sqlplus "sys/password@connect_string as sysdba" @grant_privilege.sql user_name
```

where

- `password` is the password for the `sys` user on the database.
- `connect_string` is the named SQLNet connection for this database (optional).
- `user_name` is the name of the Analytics database user.

2. Migrate the database schema.

- a) Log into the Analytics server host.
- b) Extract the database schema upgrade package (`analytics7.6_upgrade_dbschema.tar`).
- c) Navigate to the `bin` subdirectory of the extracted database schema upgrade package.
- d) Run the schema upgrade script to update the database schema and migrate the existing data.

- Windows: `schemaupgrade.bat "ANALYTICS_TOP" "schema_path\pa76upd.sql" data_tablespace index_tablespace`
- UNIX: `./schemaupgrade.sh "ANALYTICS_TOP" "schema_path\pa76upd.sql" data_tablespace index_tablespace`

where

- `ANALYTICS_TOP` is the user name on the database.
- `schema_path` is the path to the `pa76upd.sql` file. If your existing Analytics 7.5 is partitioned, this path must be to the `partitioned` directory; otherwise this path must be to the `nonpartitioned` directory.
- `data_tablespace` is the name of the tablespace where you intend to store the data. This must be the same tablespace as the previous version of Analytics.
- `index_tablespace` is the name of the tablespace where you intend to store the index. This must be the same tablespace as the previous version of Analytics.

3. Verify that you migrated the database schema successfully.

- a) From the `bin` subdirectory, verify that there were no errors in the `schemaupgrade.log` file.
- b) Verify that the `OBJECT_VERSION` columns in the `SYS_OBJECT_INFO` table all have a value of 7.6.

Optimize the Oracle database

Optional. Modify the Oracle database for optimal performance with Analytics.

1. Modify the Oracle database initialization parameters.

The following table describes the parameters you need to modify:

Parameter name	Value
<code>sga_target</code> (Oracle 10g or newer only)	Total memory × (35 to 40%)
<code>pga_aggregate_target</code>	Total memory × (35 to 40%)
<code>open_cursors</code>	300

Parameter name	Value
processes	300
db_block_size	8192
db_file_multiblock_read_count	16
optimizer_index_cost_adj	30
undo_retention	28800

If you are using Oracle 9i, you also need to modify the following parameters:

Parameter name	Value
log_buffer	1048576
sga_max_size	Total memory × (35 to 40%)
large_pool_size	104857600
shared_pool_size	314572800
java_pool_size	0
db_cache_size	sga_max_size - large_pool_size - shared_pool_size - java_pool_size - log_buffer
sort_area_size	1048576

2. Increase the size of the redo log files to decrease log file synchronization time.

Increase the number of redo log groups to four and increase the size of each of the redo log groups to 1.5 GB.

In addition, you should redistribute these redo log files to a physical hard disk other than the one containing the operating system and the tablespaces for Analytics data.

3. Increase the UNDO and TEMP tablespaces.

Change the size of the UNDO and TEMP tablespaces to 20 GB, or as large as possible.

For example, if your cluster has a large volume of records (such as a full workload scale LSF cluster consisting of 5000 hosts and 3 million finished jobs a day with 40 000 license usage combinations), you may need to set the TEMP and UNDO tablespace size to 40 GB or more.

Verify the bjobs data loader deployment

Analytics 7.6 includes enhancements to data transformers for the bjobs data loader (lsfjobobsloader). You need to verify that you already deployed the bjobs data loader to your Analytics 7.5 installation prior to installing Analytics 7.6

1. Verify that the bjobs data loader is already deployed to Analytics.

- a) Log into the Analytics server host.
- b) View or edit the `ANALYTICS_TOP/tasks/LSFDataAggHourly/lsfdataagghourly.task` file.
- c) Look for the following lines:

```
# lsfGenJobInfo - COVERED BY BJOBS SOLUTION. TO BE REMOVED
```

```
# lsfJobStatisticsHourly - COVERED BY BJOBS SOLUTION. TO BE REMOVED  
# lsfJobResUsageByGroupHourly - COVERED BY BJOBS SOLUTION. TO BE REMOVED  
# lsfJobSlotsUsageByGroupHourly - COVERED BY BJOBS SOLUTION. TO BE REMOVED
```

If the lines are present in the file, you have not deployed the bjobs data loader and need to deploy it prior to installing Analytics 7.6.

Note:

Instead of searching for the entire lines, you may search for just the names to verify that they have been commented out. That is, search for # lsfGenJobInfo, # lsfJobStatisticsHourly, # lsfJobResUsageByGroupHourly, and # lsfJobSlotsUsageByGroupHourly.

2. If required, deploy the bjobs data loader to Analytics.

You need to deploy bjobs- sp012 to Analytics. Contact Platform Computing to obtain the necessary files.

Prepare for the Analytics upgrade

Upgrade Analytics

Upgrade Analytics from version 7.5 to version 7.6.

What you need to do

Check the following:

- The Analytics server and node hosts are running Analytics 7.5.
The Analytics upgrade packages will work with any patches or solution packages that you installed, but will only work on version 7.5.
- The Analytics server and node hosts meet the detailed system and software requirements for Analytics 7.6.

Refer to the *Release Notes for Platform Analytics* for the latest requirements.

1. [Upgrade the Analytics server](#) on page 14
2. [Verify the Analytics server upgrade](#) on page 17
3. [Upgrade the Analytics node](#) on page 18
4. [Verify the Analytics node upgrade](#) on page 21

After completing the Analytics upgrade, you can optionally optimize the Analytics node and server hosts.

Upgrade the Analytics server

Upgrade the Analytics server on a UNIX or Windows host.

Check the following:

- You migrated the Oracle database schema to Analytics 7.6.

Important:

The Analytics upgrade package cannot detect whether you updated the database schema. If you proceed with the upgrade without migrating the database schema, some Analytics server tasks will encounter problems and send events.

- If you are upgrading on a UNIX host, your host must have access to the X-Windows environment:
 - If you are connected to the Analytics server host locally, the host must be running X-Windows.
 - If you are connected to the Analytics server host via `tel net`, you must be running `xserver` on your local host.

Upgrade the Analytics server using the Analytics server upgrade package. The upgrade package will make backups of the previous version of Analytics before upgrading your server to the latest version.

1. Log into the Analytics server host with the same account that you used to install the previous version of Analytics.
2. If you connected to the UNIX host via `tel net` and are running `xserver` on a local host, set your display environment.

Test your display by running `xclock` or another X-Windows application.

If the application displays, your display environment is already set correctly; otherwise, you need to set your display environment.

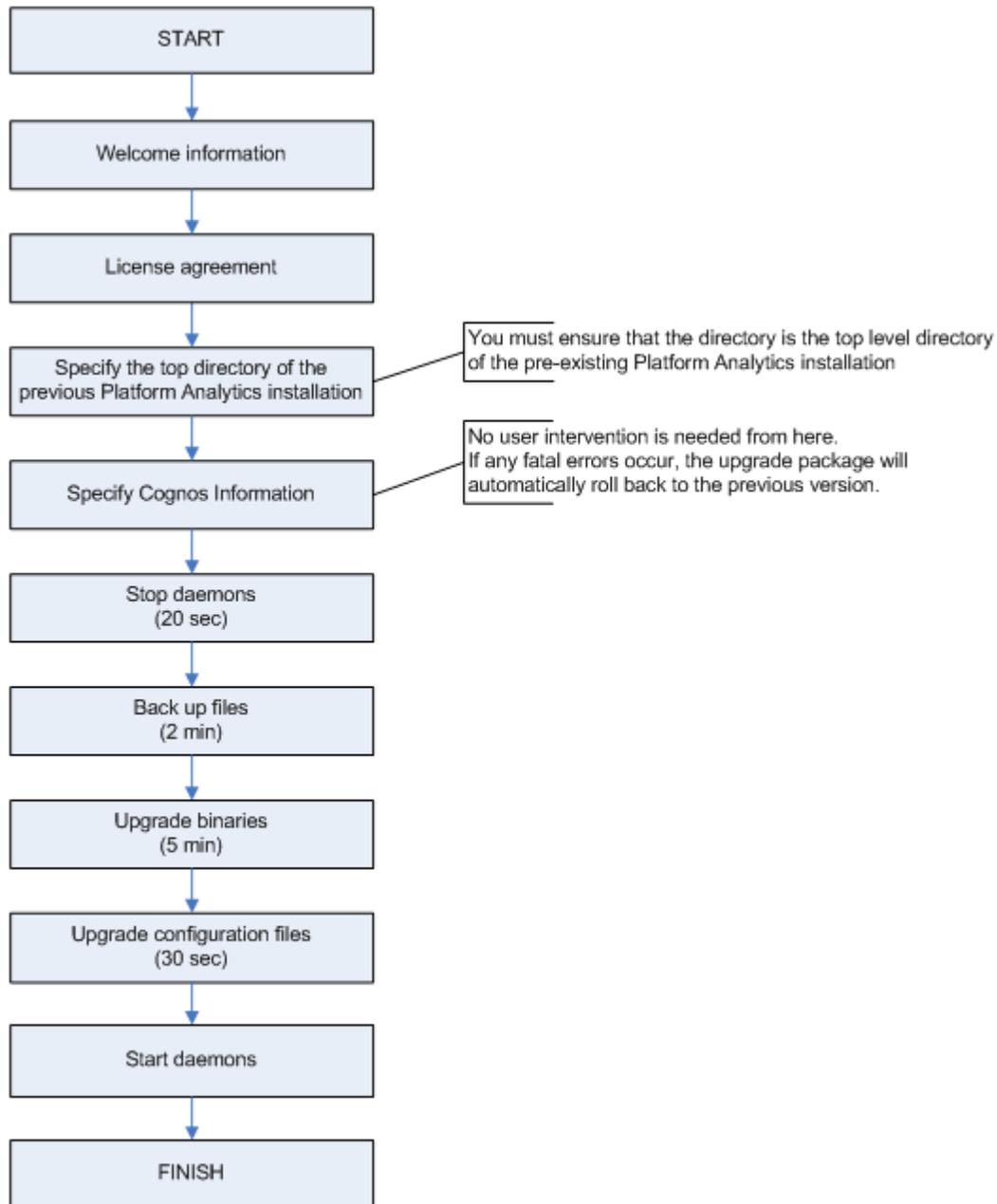
- For `csh` or `tcsh`:
setenv DISPLAY hostname:0.0

- For sh, ksh, or bash:
DISPLAY=hostname:0.0
export DISPLAY

where *hostname* is your local host.

3. Run the Analytics server upgrade package.
 - UNIX: ***JAnalyticsversion_server_upgrade_platform.bin***
 - Windows: ***Analyticsversion_server_upgrade_platform.exe***
4. Follow the dialog prompts to complete the upgrade.

The following diagram illustrates the dialog for the Analytics server upgrade package:



After completing the dialog prompts, the upgrade package upgrades the Analytics server and the cubes on the Cognos server.

Important:

If the Cognos server is down when the upgrade package upgrades the Analytics and Cognos server components, you will encounter the following error message:

"Setup failed to connect to the Cognos PowerPlay Enterprise Server."

Take note of this error, because you will need to publish the cubes manually after completing the upgrade.

5. If you are upgrading on a UNIX host, start the Analytics server.

Run **daemons.sh start** from the *ANALYTICS_TOP/bin* directory.

6. Add new cubes to the build cubes scheduled tasks.

The Analytics 7.6 upgrade package introduces new cubes, which are not automatically built. To ensure that these cubes are automatically built with the previous cubes, add the new cubes to the build cubes scheduled tasks using the Analytics Console.

- a) Start the Analytics Console.

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

- b) Click Scheduled Tasks in the navigation tree.

- c) In the main window, right-click BuildCubes and select Edit Scheduled Task.

- d) Move the new daily cubes from the Disabled column to the Enabled column.

The new daily cubes for Analytics 7.6 are as follows:

- Cluster Capacity
- Daily Hardware (with Boolean Resources and Core Utilization)
- Daily Hardware (with Core Utilization)
- Daily License Usage Statistics
- Daily License Usage Statistics by Feature
- Hardware Histogram
- Health Check
- Jobs vs Slot Utilization
- License Usage by Feature Histogram
- License Usage by Server Histogram

- e) In the main window, right-click BuildCubesHourly and select Edit Scheduled Task.

- f) Move the new hourly cubes from the Disabled column to the Enabled column.

The new hourly cubes for Analytics 7.6 are as follows:

- Hourly Hardware (with Boolean Resources and Core Utilization)
- Hourly Hardware (with Core Utilization)
- Hourly License Usage Statistics
- Hourly License Usage Statistics by Feature

The changed files from your previous version of Analytics are backed up to the following location:

- UNIX: *ANALYTICS_TOP/upgrade_backup/version_number*
- Windows: *ANALYTICS_TOP\upgrade_backup\version_number*

If you encountered a problem with connecting to the Cognos PowerPlay Enterprise Server, you will need to start the Cognos server and manually publish all the cubes.

Verify that you have successfully upgraded the Analytics server.

Verify the Analytics server upgrade

Verify that you have successfully upgraded the Analytics server on a UNIX or Windows host.

You upgraded the Analytics server and are still logged into the host.

1. Verify that there are no errors generated in the event log file after the upgrade.

The event log file is named `event.1og.host_name` and is located in the PERF log directory:

- UNIX: `ANALYTICS_TOP/1og`
- Windows: `ANALYTICS_TOP\1og`

2. Start the Analytics Console.

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

3. Click Scheduled Tasks in the navigation tree and verify that there are no failed tasks.

4. After at least an hour, verify that the relevant cubes are successfully upgraded.

You need to wait at least one hour for Analytics to perform aggregation at least once after the upgrade.

- a) Access the Cognos server web interface by entering the `/cognos` subdirectory of your Analytics server domain name in your browser.

For example, if your Analytics server host is `hostA` in the domain `example.com`, enter `http://hostA.domain.example.com/cognos` in your browser. If this does not work, try using the web server port number. For the default port of 80, try `http://hostA.domain.example.com:80/cognos`.

- b) In the navigation tree, select NewsIndex > Platform Analytics > Cubes.
- c) Click Cluster Capacity cube and verify the upgrade.

The cube should open without errors.

- d) Click Return to Source, then click Jobs vs. Slot Utilization cube and verify the upgrade.

The cube should open without errors.

- e) Open various cube views and verify that the data is correct.

5. If your Analytics installation has customizations, check that the customizations are still present.

Upgrade the Analytics node

Upgrade the Analytics node on a UNIX or Windows host.

Check the following:

- You migrated the Oracle database schema to Analytics 7.6.

Important:

The Analytics upgrade package cannot detect whether you updated the database schema. If you proceed with the upgrade without migrating the database schema, the Analytics data loaders will encounter problems and send events.

- If you are upgrading on a UNIX host, your host must have access to the X-Windows environment:
 - If you are connected to the intended Analytics node host locally, the host must be running X-Windows.
 - If you are connected to the intended Analytics node host via telnet, you must be running xserver on your local host.

Upgrade the Analytics node using the Analytics node upgrade package. The upgrade package will make backups of the previous version of Analytics before upgrading your node to the latest version.

The upgrade package stops the Analytics data loaders to upgrade these components. These data loaders will not be stopped for longer than five minutes under normal circumstances.

1. Log into the Analytics node host with the same account that you used to install the previous version of Analytics.
2. Determine if there are multiple loader controllers running on the Analytics node.

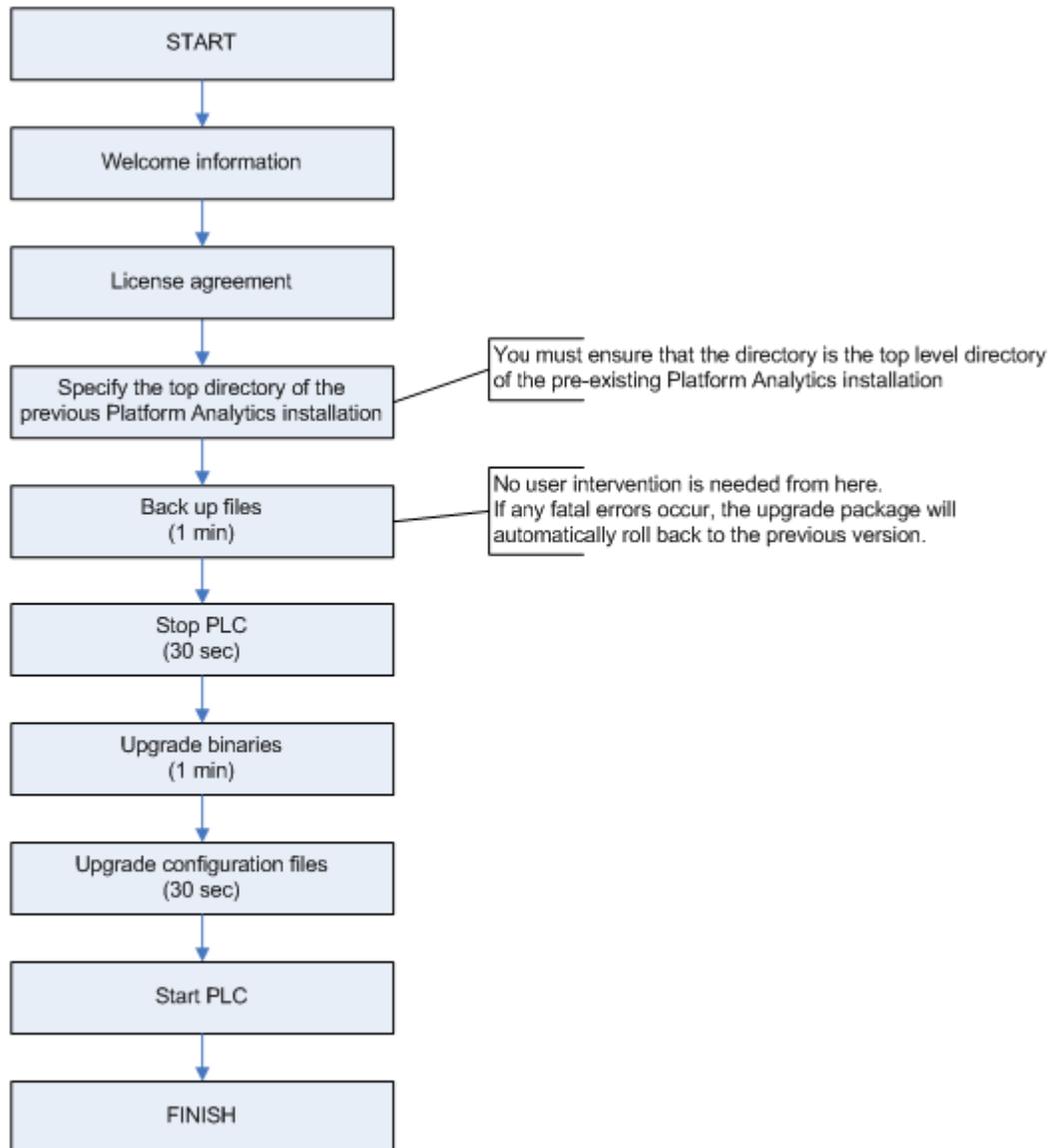
perfadmin list

3. If the Analytics node is running multiple loader controllers, manually shut down all loader controllers except for the one named "pl c", which is the original loader controller service.

perfadmin stop loader_controller_name

4. Run the Analytics node upgrade package.
 - UNIX: ***JAnalyticsversion_node_upgrade_platform.bin***
 - Windows: ***Analyticsversion_node_upgrade_platform.exe***
5. Follow the dialog prompts to complete the upgrade.

The following diagram illustrates the dialog for the Analytics node upgrade package:



6. If you are upgrading on a UNIX host, source the PERF environment.

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

If you want to start the Analytics node later, you will need to source the LSF environment before starting the Analytics node.

7. Verify that the `plc` service has started.

Run **perfadmin list** and verify that `plc` is running.

8. If the `plc` service has not already started, manually start the `plc` service.

perfadmin start plc

9. If the Analytics node is running multiple loader controllers, manually start all loader controllers except for "`plc`", which is the original loader controller service.

perfadmin start loader_controller_name

10. Verify that you upgraded the Analytics node successfully.

a) Verify that the data loaders are running.

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

b) Verify that there are no errors in the loader controller log file.

The loader controller log file is named `plc.l og. host_name` and is located in the PERF log directory:

- UNIX: `ANALYTICS_TOP/l og`
- Windows: `ANALYTICS_TOP\l og`

c) After at least one hour, verify that there are no errors in any of the data loader log files.

You need to wait at least one hour because this is the longest data sampling interval of the data loaders.

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

- UNIX: `ANALYTICS_TOP/l og/data loader`
- Windows: `ANALYTICS_TOP\l og\data loader`

The changed files from your previous version of Analytics are backed up to the following location:

- UNIX: `ANALYTICS_TOP/upgrade_backup/versi on_number`
- Windows: `ANALYTICS_TOP\upgrade_backup\versi on_number`

Verify the Analytics node upgrade

Verify that you have successfully upgraded the Analytics node on a UNIX or Windows host.

You upgraded the Analytics node and are still logged into the host.

1. Verify that the data loaders are running.

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

2. After at least one hour, verify that there are no errors in any of the data loader log files.

You need to wait at least one hour because this is the longest data sampling interval of the data loaders.

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

- UNIX: `ANALYTICS_TOP/l og/data loader`
- Windows: `ANALYTICS_TOP\l og\data loader`

3. Verify that there are no errors in the loader controller log file.

The loader controller log file is named `plc.l og. host_name` and is located in the PERF log directory:

- UNIX: `ANALYTICS_TOP/l og`
- Windows: `ANALYTICS_TOP\l og`

Optimize the Analytics node

Modify the Analytics node to enhance performance.

What you need to do

The following steps are optional.

1. [Increase JVM memory](#) on page 24
2. [Disperse the Analytics node workload](#) on page 24
3. [Optimize specific data loaders](#) on page 27

After optimizing the Analytics node, you can optionally optimize the Analytics server.

Increase JVM memory

Increase the Java Virtual Machine (JVM) memory to at least 2 GB for the Analytics node instead of using the default value to prevent the JVM from running out of memory.

Note:

Java cannot use more than 1638 MB of memory on 32-bit platforms. If you are using 32-bit Windows, you can only increase the JVM memory to 1638 MB.

1. Log into the Analytics node host.
2. Edit the `ANALYTICS_TOP/conf/wsm/wsm_plc.conf` file.
3. Change the Java starting options to increase the JVM memory.

Navigate to `JAVA_OPTS` and increase the JVM memory to at least 2 GB.

For example,

```
JAVA_OPTS=-Xms64m -Xmx2048m
```

Disperse the Analytics node workload

You can reduce performance issues by dispersing the Analytics node workload among multiple hosts or multiple loader controllers within a host so that each host or loader controller is responsible for a specific type (or types) of data loading.

Based on the type of data, you can categorize all the data loaders into different types, and enable each node or loader controller to be responsible for only one type of data loader. The following table describes the different data loader categories and their corresponding loader controller configuration files:

Data loader category	Loader controller configuration file
Host-related	<ul style="list-style-type: none">• <code>plc_ego.xml</code>• <code>plc_coreutil.xml</code>
Job-related	<ul style="list-style-type: none">• <code>plc_lsf.xml</code>• <code>plc_bj_obs-sp012.xml</code>
Advanced job-related	<ul style="list-style-type: none">• <code>plc_lsf_advanced.xml</code>

Data loader category	Loader controller configuration file
License-related	<ul style="list-style-type: none"> <code>pl c_l i c e n s e. x m l</code>

- [Disperse the Analytics node workload to multiple hosts](#) on page 25
 Disperse the workload to multiple hosts if your Analytics node is experiencing performance issues and cannot handle all the workload by itself.
- [Disperse the Analytics node workload to multiple loader controllers in one node](#) on page 25
 Disperse the workload to multiple loader controllers within a host to reduce performance bottlenecks if your Analytics node host is powerful enough to handle the workload (for example, the host has at least four cores and at least 6 GB of memory).

Disperse the Analytics node workload to multiple hosts

A single node might experience performance issues when handling the high workload. To resolve this issue, you can disperse the node workload to multiple hosts so that each host is responsible for a specific type of data loading.

1. Install and configure the Analytics node to other hosts in the same LSF cluster.

For example, if you plan to have one node for each data loader category, you should install and configure three additional Analytics node hosts. Therefore, your original node would handle host-related data loaders, while each of the other three nodes would handle each of the other three categories.

Alternatively, you can have fewer nodes and still disperse workload to some degree. For example, you could have two nodes: your original node could handle host-related and job-related data loaders while another node could handle advanced job-related and license-related data loaders.

2. For each Analytics node, disable the data loading categories that the node will not be handling.
 - a) Navigate to the `ANALYTICS_TOP/conf/pl c` directory.
 - b) Move or rename the configuration files for the data loader categories that the Analytics node will no longer handle.

For example, if you want the Analytics node to only handle host-related data loaders, move or rename all configuration files except `pl c_ego. xml` and `pl c_coreuti l . xml`.

Disperse the Analytics node workload to multiple loader controllers in one node

If your node host is powerful enough to handle the workload (for example, the host has at least four cores and at least 6 GB of memory), you can reduce the performance bottlenecks by creating multiple loader controllers and having each loader controller be responsible for a specific type of data loading. This enables each of the multiple cores in your host to control a single loader controller and be more efficient in sharing the workload among multiple cores.

1. For each data loader category beyond the first, create a new loader controller on the same host.

Therefore, since there are four data loader categories, create three additional loader controllers.

In this section, `PLC_NAME` represents the name of the new loader controller that you will create.

For example, you can create `pl c_j o b` to handle job-related data loaders, `pl c_advanced_j o b` to handle advanced job-related data loaders, and `pl c_l i c e n s e` to handle license-related data loaders, while the original loader controller (`pl c`) handles host-related data loaders.

- a) From the *ANALYTICS_TOP/conf/wsm* directory, copy the *wsm_plc.conf* file to a new file, one for each new loader controller.

For each new loader controller, name the new file *wsm_PLC_NAME.conf*

For example, copy *wsm_plc.conf* to *wsm_plc_job.conf*, *wsm_plc_advanced_job.conf*, and *wsm_plc_license.conf*.

- b) Edit each new *wsm_PLC_NAME.conf* file and specify the new loader controller name.

Navigate to the *SERVICE_NAME*, *SERVICE_COMMAND*, and *LOG_PREFIX* parameters to use the new loader controller name as follows:

```
SERVICE_NAME=PLC_NAME
SERVICE_COMMAND=com.platform.perf.dataloader.Main -f PLC_NAME.xml
LOG_PREFIX=PLC_NAME
```

For example, for the *wsm_plc_job.conf* file:

```
SERVICE_NAME=plc_job
SERVICE_COMMAND=com.platform.perf.dataloader.Main -f plc_job.xml
LOG_PREFIX=plc_job
```

Make similar edits to the *wsm_plc_advanced_job.conf* and *wsm_plc_license.conf* files.

- c) From the *ANALYTICS_TOP/conf* directory, create a new subdirectory for each new loader controller.

Name each directory *PLC_NAME*.

For example, create three new directories named *plc_job*, *plc_advanced_job*, and *plc_license*.

- d) Move the loader controller configuration files, corresponding to the data loader categories, from *ANALYTICS_TOP/conf/plc* to the subdirectory corresponding to the new loader controller that will handle the category data loaders.

For example,

1. Move *plc_lsf.xml* and *plc_bj_obs-sp012.xml* from *ANALYTICS_TOP/conf/plc* to *ANALYTICS_TOP/conf/plc_job*
2. Move *plc_lsf_advanced.xml* from *ANALYTICS_TOP/conf/plc* to *ANALYTICS_TOP/conf/plc_advanced_job*.
3. Move *plc_license.xml* from *ANALYTICS_TOP/conf/plc* to *ANALYTICS_TOP/conf/plc_license*.

- e) From the *ANALYTICS_TOP/conf* directory, copy the loader controller file (*plc.xml*) to a new file, one for each new loader controller.

For each new loader controller, name the new file *PLC_NAME.xml*

For example, copy *plc.xml* to *plc_job.xml*, *plc_advanced_job.xml*, and *plc_license.xml*.

- f) Edit each new loader controller file and change the *Port* parameter to a new port and change the *PLCDir* parameter to the new loader controller directory.

For example,

1. In *plc_job.xml*, change *Port* to from **4046** to **4047** and change *PLCDir* from **plc** to **plc_job**.
2. In *plc_advanced_job.xml*, change *Port* to from **4046** to **4048** and change *PLCDir* from **plc** to **plc_advanced_job**.

3. In `plc_license.xml`, change `Port` to from **4046** to **4049** and change `PLCDir` from `plc` to `plc_license`.
2. Restart the new loader controllers.

```
perfadmin stop all
```

```
perfadmin start all
```

Note:

To stop or start an individual data loader, use `perfadmin stop PLC_NAME` and `perfadmin start PLC_NAME`

For example, to stop the loader controller that handles license-related data loaders, run `perfadmin stop plc_license`.

Optimize specific data loaders

If you are encountering specific problems or are working under specific environments, you can optimize certain data loaders to enhance the working performance of these individual data loaders.

1. Optimize the FLEXnet usage data loader (`f1exliscusageloader`) to improve data loading time if it cannot finish loading the data within one sampling interval.

If the FLEXnet usage data loader cannot finish the data loading of one sampling interval within the time of one sampling interval (typically five minutes), the data loader will be behind. You need to enable the multi-threads setting to catch up with the multi-servers and daemons workload scale.

- a) Log into the Analytics node host that is running the FLEXnet usage data loader.
- b) Edit the `ANALYTICS_TOP/conf/dataloader/f1exliscusage.properties` file.
- c) Edit the `ThreadNumber` parameter to increase the number of threads for collecting data concurrently.

For example,

```
ThreadNumber=10
```

2. Optimize the FLEXnet Manager data loader (`fnmloader`) to achieve a larger capacity if you need to handle more than 1 million events per day.

Using the default configuration means that you do not have to modify the data schema of the FLEXnet Manager reporting database, but the capacity of this data loader is limited to 1 million events per day.

If you need to achieve a larger capacity, you need to modify the data schema of the FLEXnet Manager reporting database to increase performance. For more details, refer to the `ANALYTICS_TOP/conf/dataloader/FNMLoader.readme` file.

3. Optimize the LSF events data loader (`lsfeventsloader`) to prevent data loss if you are using LSF versions 7.0.3 or earlier.

The LSF events data loader reads data from the `lsb.stream` file. If you are using LSF versions 7.0.3 or earlier, you should increase the size of the `lsb.stream` file to 2 GB or more to prevent data loss.

- a) Log into the Analytics node host that is running the LSF events data loader.
- b) Edit the `LSF_ENVDIR/lsbatch/cluster_name/configdir/lsb.params` file.
- c) In the `Parameters` section, define a new parameter named `MAX_EVENT_STREAM_SIZE`.

For example,

```
Begin Parameters
...
MAX_EVENT_STREAM_SIZE = 2048
...
```

Optimize the Analytics node

End Parameters

- d) Reconfigure the LSF cluster for your changes to take effect.

lsadmin reconfig

Optimize the Analytics server

Modify the Analytics server to enhance performance.

What you need to do

The following steps are optional.

1. [Modify the Cognos parameters](#) on page 30
2. [Increase JVM memory](#) on page 31
3. [Improve the hourly build cubes task to show data from the last hour](#) on page 32
4. [Split the data transformer tasks to disperse workload](#) on page 33
5. [Optimize the data purger \(non-partitioned data schema only\)](#) on page 39
6. [Change the data retention period](#) on page 40

Modify the Cognos parameters

Edit the Cognos parameters to enhance the performance of the Cognos server and to improve cube building performance. The parameters you edit depend on the operating system of the Analytics server host.

- [Modify the Cognos parameters in a Windows host](#) on page 30
- [Modify the Cognos parameters in a Solaris host](#) on page 30

Modify the Cognos parameters in a Windows host

1. Log into the Analytics server host.
2. Navigate to the `cer5\bin` subdirectory of the Cognos installation directory.

If you installed Cognos to the default installation directory, navigate to `C:\Program Files\Cognos\cer5\bin`.

3. Edit the `cer5.ini` file and change the following parameter definitions:

```
WriteCacheSize=65536  
SortMemory=10240
```

4. Edit the `trnsfrmr.ini` file (or create a new file if it does not exist), and change (or add) the following parameter definitions:

```
[PowerPlay Transformer]  
MaxTransactionNum=800000  
MultiFileCubeThreshold=1000000
```

Modify the Cognos parameters in a Solaris host

1. Log into the Analytics server host.
2. Navigate to the `cer5/bin` subdirectory of the Cognos installation directory.

If you installed Cognos to the default installation directory, navigate to `/usr/cognos/cer5/bin`.

3. Edit the `cer5.ini` file and change the following parameter definitions:

```
SORT_MEMORY=10240
```

4. Edit the `trnsfrmr.rc` file (or create a new file if it does not exist), and change (or add) the following parameter definitions:

```
[PowerPlay Transformer]  
MaxTransactionNum=800000  
MultiFileCubeThreshold=1000000
```

- Optional. Edit *ANALYTICS_TOP/bin/cr_cube.sh* and change the *PPDS_WRITE_MEMORY* environment variable definition to another value.

The default parameter definition (65536) is already at the optimal value, so you should not need to change this value.

Increase JVM memory

Increase the Java Virtual Machine (JVM) memory for the data transformer and cube and datamart building. The file and parameters that you modify depend on the operating system of the Analytics server host.

- [Increase JVM memory in a Windows host](#) on page 31
- [Increase JVM memory in a Solaris host](#) on page 32

Increase JVM memory in a Windows host

Note:

Java cannot use more than 1638 MB of memory on 32-bit platforms. If you are using 32-bit Windows, you can only increase the JVM memory to 1638 MB.

- Log into the Analytics server host.
- Increase the JVM memory for the data transformer.

Edit the *ANALYTICS_TOP/bin/runETL.bat* file and change the Java starting command line to the following:

- Windows 32-bit (you can only change this to a maximum of 1638 MB):

```
"%JAVA_HOME%\bin\java.exe" -Xms64m -Xmx1638m
```
- Windows 64-bit (you can change this to 2 GB or more):

```
"%JAVA_HOME%\bin\java.exe" -Xms64m -Xmx2048m
```

Note:

If you have a large cluster (with at least 1000 hosts), you should increase the JVM memory to at least 4 GB. Therefore, you must be using a Windows 64-bit host, and you should change the Java starting command to use at least 4096 MB:

```
"%JAVA_HOME%\bin\java.exe" -Xms64m -Xmx4096m
```

- Increase the JVM memory for cube and datamart building.

Edit the *ANALYTICS_TOP/bin/rundatamart.bat* and change the Java starting command line to the following:

- Windows 32-bit (you can only change this to a maximum of 1638 MB):

```
"%JAVA_HOME%\bin\java.exe" -Xms64m -Xmx1638m
```
- Windows 64-bit (you can change this to 2 GB or more):

```
"%JAVA_HOME%\bin\java.exe" -Xms64m -Xmx2048m
```

Increase JVM memory in a Solaris host

1. Log into the Analytics server host.
2. Increase the JVM memory for the data transformer.

Edit the `ANALYTICS_TOP/bin/ex_runETL.sh` file and change the Java starting command line to use 2 GB or more memory, as follows:

```
$JAVA_HOME/bin/java -Xms64m -Xmx2048m
```

Note:

If you have a large cluster (with at least 1000 hosts), you should increase the JVM memory to at least 4 GB. Therefore, you should change the Java starting command to use at least 4096 MB:

```
$JAVA_HOME/bin/java -Xms64m -Xmx4096m
```

3. Increase the JVM memory for the data transformer.

Edit the `ANALYTICS_TOP/bin/rundatamart.sh` file and change the Java starting command line to use 2 GB or more memory, as follows:

```
$JAVA_HOME/bin/java -Xms64m -Xmx2048m
```

Improve the hourly build cubes task to show data from the last hour

By default, the scheduled task for building hourly OLAP cubes is scheduled to run every day. The cubes are published to Cognos Upfront after they are successfully built. Since the scheduled task runs every day, the hourly cubes only show the data from the last day. If you want the hourly cubes to always show data from the last hour, you should separate hourly cubes from the build cubes task into new scheduled tasks, and configure the Analytics server to run these tasks in parallel.

Before performing this optimization, you must determine how many cubes your Analytics server can run in parallel, which you can calculate by using the number of cores available in your host:

$(\text{Number of parallel build cubes}) = (\text{Number of cores available}) - 4$

The four cores that you subtract from the calculation are due to the four cores that are assigned to other Analytics tasks. Therefore, you cannot build two cubes in parallel unless the Analytics server has six or more cores available. To obtain a one hour turnaround time for all hourly cubes, the recommended value for this parameter will vary.

For example, in a cluster with 2 000 hosts and 1 000 000 finished jobs per day, you need to build at least four cubes in parallel, which means you need to have at least eight cores available. In a cluster with 5 000 hosts and 3 000 000 finished jobs per day, you need to build at least five cubes in parallel, which means you need to have at least nine cores available.

1. Log into the Analytics server host.
2. Launch the Analytics Console.
 - UNIX: `ANALYTICS_TOP/bin/runconsole.sh`
 - Windows: Start > Programs > Platform Analytics Server > Platform Analytics Console
3. For each cube that the Analytics server will build in parallel, create a new scheduled task to run every hour starting from 2:00 a.m.

For example, to build four cubes in parallel, repeat the following steps to create four scheduled tasks named `Bui l dCubesHour l y1`, `Bui l dCubesHour l y2`, `Bui l dCubesHour l y3`, and `Bui l dCubesHour l y4`.

- a) Click Scheduled Tasks in the navigation tree.
- b) Right-click the main window and select Add Scheduled Task.
- c) Complete the required fields for the new task.
 - Scheduled Task: Specify the name of this task.
 - Script File: Specify **bin/buildcubes.js**.
 - Script Function: Specify **doit**.
- d) Enable the new scheduled task.
- e) Change the new scheduled task to start at 2:00 a.m.
- f) Select the Run every: field and specify 1 hour.
- g) In the Cubes panel, select the hourly cubes you want to build with this task and move them from Disabled to Enabled.

Tip:

Distribute the hourly cubes evenly among all the new scheduled tasks.

- h) To save your changes and close the window, click OK.
4. From the `Bui l dCubesHour l y` scheduled task, disable the hourly cubes that the new scheduled tasks will build.
 - a) Click Scheduled Tasks in the navigation tree.
 - b) Right-click the `Bui l dCubesHour l y` scheduled task in the main window and select Edit Scheduled Task.

The Scheduled Task window for `Bui l dCubesHour l y` displays.
 - c) In the Cubes panel, select the hourly cubes that the new scheduled tasks will build move them from Enabled to Disabled.
 5. Exit from the Analytics Console.
 6. Edit the `pi . conf` file.
 - UNIX: `ANALYTICS_TOP/conf/pi . conf`
 - Windows: `ANALYTICS_TOP\conf\pi . conf`
 7. Increase the specified value for the `Bat chBui l dThreshol d` parameter to the number of cubes that the Analytics server will build in parallel.

This parameter, which can have a value from 1 to 6, specifies the number of cubes that can be built in parallel.

For example, to build four cubes in parallel,

BatchBuildThreshold = 4

Split the data transformer tasks to disperse workload

By default, there are four default scheduled tasks that control data transformers. Four scheduled tasks might not be enough to be able to run all the data transformers within one hour, so to enhance performance, you can split these data transformers into more tasks.

The following is a recommended format for splitting your data transformers into seven tasks. The examples will make use of this table (specifically, with Task 2). In the following example, Tasks 1 to 5 are hourly scheduled tasks, while Tasks 6 to 7 are daily scheduled tasks.

Task	Data transformer name	Data flow entry
1	IsfJobResUsageByGroupHourly	datatransformer/flow/bjobs-sp012/hourly/jobdetailresusage/main_job_resusage_bygroup.xml
2	IsfJobSlotsUsageByGroupHourly	datatransformer/flow/bjobs-sp012/hourly/jobdetailresusage/main_job_slotsusage_bygroup.xml
	IsfJobStatisticsHourly	datatransformer/flow/bjobs-sp012/hourly/jobdetailresusage/main_job_slotsusage_bygroup.xml
3	IsfJobThroughputHourly	datatransformer/flow/jobrelateddataaggregation/hourly/jobthroughput/main_job_throughput_etl.xml
	IsfJobmartHourly	datatransformer/flow/jobrelateddataaggregation/hourly/jobmart/main_jobmart_etl.xml
	IsfJobExecProcHourly	datatransformer/flow/jobrelateddataaggregation/hourly/jobexecproc/main_job_execproc_etl.xml
	flexEventHourly	datatransformer/flow/flexnetevents/hourly/main_flexEventHourly.xml
	flexEventDenialHourly	datatransformer/flow/flexnetevents/hourly/flexeventdenialhourly/main_flexEventDenialHourly.xml
	flexEventDenialByFeatureHourly	datatransformer/flow/flexnetevents/hourly/flexeventdenialhourly/main_flexEventDenialByFeatureHourly.xml

Task	Data transformer name	Data flow entry
4	hostPropertiesHourly	datatransformer/flow/hostproperties/hourly/ main_host_properties.xml
	hostMetricsHourly	datatransformer/flow/hostmetrics/hourly/ main_host_metrics_hourly.xml
	hostGroupConfHourly	datatransformer/flow/lsfhostgroupconf/hourly/ main_host_group_conf_hourly.xml
	userGroupConfHourly	datatransformer/flow/lsfusergroupconf/hourly/ main_userGroupConfHourly.xml
	lsfSharedResourceUsageHourly	datatransformer/flow/lsfsharedresourceusage/hourly/ main_sharedResourceUsageHourly.xml
	lsfSharedResourcePropertiesHourly	datatransformer/flow/lsfsharedresourceproperties/hourly/ main_sharedResourceProperties.xml
	lsfHostStatusHourly	datatransformer/flow/lsfhoststatus/hourly/ main_lsf_host_status_hourly.xml
	lsfJobExecProcHourly	datatransformer/flow/jobrelateddataaggregation/hourly/ jobexecproc/main_job_execproc_etl.xml
	lsfJobmartHourly	datatransformer/flow/jobrelateddataaggregation/hourly/jobmart/ main_jobmart_etl.xml
	lsfJobThroughputHourly	datatransformer/flow/jobrelateddataaggregation/hourly/ jobthroughput/main_job_throughput_etl.xml
	lsfJobSlotUsageByHost	datatransformer/flow/jobrelateddataaggregation/hourly/ jobslotusagebyhost/main_jobslotusagebyhost_etl.xml
	lsfJobSlotUsageByQueue	datatransformer/flow/jobrelateddataaggregation/hourly/ jobslotusagebyqueue/main_jobslotusagebyqueue_etl.xml
	lsfJobPendingReasonHourly	datatransformer/flow/lsfpendingreason/hourly/ main_job_pending_reason_hourly.xml
	lsfDetailPendingReasonHourly	datatransformer/flow/lsfdetailpendreason/hourly/hostreason/ main_detail_pending_reason_hourly.xml
	jobs_v_slotutil	datatransformer/flow/jobrelateddataaggregation/hourly/ job_v_slotutil/main_job_v_slotutil_etl.xml
	cluster_slots	datatransformer/flow/jobrelateddataaggregation/hourly/ job_v_slotutil/main_cluster_slots_etl.xml
	hostMetricsHistogram	datatransformer/flow/hardware_and_license_histogram-sp011/ hostmetrics_histogram/main_host_metrics_hourly.xml
	hardwareCoreUtilHourly	datatransformer/flow/hardware_with_core_util-sp006/hourly/ main_hardware_core_util_hourly.xml

Optimize the Analytics server

Task	Data transformer name	Data flow entry
5	licHourly	datatransformer/flow/flexlmusage/hourly/ main_license_usage_hourly.xml
	licUtilByFeature	datatransformer/flow/flexlmusage/hourly/ main_license_usage_hourly_byfeature.xml
	licDenialHourly	datatransformer/flow/flexlmevents/hourly/licevents hourly/ main_license_denial_hourly.xml
	licDenialByFeatureHourly	datatransformer/flow/flexlmevents/hourly/licevents hourly/ main_license_denial_hourly_byfeature.xml
	licenseUsageByFeatureHistogram	datatransformer/flow/hardware_and_license_histogram-sp011/ licusage_histogram/main_license_usage_hourly_byfeature.xml
	licenseUsageByServerHistogram	datatransformer/flow/hardware_and_license_histogram-sp011/ licusage_histogram/main_license_usage_hourly.xml

Task	Data transformer name	Data flow entry
6	hostMetricsDaily	datatransformer/flow/hostmetrics/daily/ main_host_metrics_daily.xml
	hostNotBuiltinMetricsDaily	datatransformer/flow/hostmetrics/daily/ main_host_notbuiltin_metrics_daily.xml
	hostGroupConfDaily	datatransformer/flow/lsfhostgroupconf/daily/ main_host_group_conf_daily.xml
	IsfHostStatusDaily	datatransformer/flow/lsfhoststatus/daily/ main_lsf_host_status_daily.xml
	IsfSharedResourcesDaily	datatransformer/flow/lsfsharedresourceusage/daily/ main_sharedResourceUsageDaily.xml
	IsfJobmartDaily	datatransformer/flow/jobrelateddataaggregation/daily/jobmart/ main_jobmart_etl_daily.xml
	IsfJobmartHPCDaily	datatransformer/flow/jobrelateddataaggregation/daily/jobmart/ main_hpc_jobmart_etl_daily.xml
	IsfJobResUsageCostDaily	datatransformer/flow/jobrelateddataaggregation/daily/ jobresusagecost/main_jobresusagecost_etl_daily.xml
	IsfJobThroughputDaily	datatransformer/flow/jobrelateddataaggregation/daily/ jobthroughput/main_jobthroughput_etl_daily.xml
	IsfJobStatisticsDaily	datatransformer/flow/jobrelateddataaggregation/daily/ jobstatistics/main_jobnumberofjobs_etl_daily.xml
	IsfJobResUsageByGroupDaily	datatransformer/flow/jobrelateddataaggregation/daily/ jobresusagebygroup/main_jobresusagebygroup_etl_daily.xml
	IsfJobSlotUsageByGroupDaily	datatransformer/flow/jobrelateddataaggregation/daily/ jobslotsusagebygroup/main_jobslotusagebygroup_etl_daily.xml
	IsfJobPendingReasonDaily	datatransformer/flow/lsfpendingreason/daily/ main_job_pending_reason_daily.xml
	IsfDetailPendingReasonJobNum Daily	datatransformer/flow/lsfdetailpendreason/daily/hostreason/ jobnumber/main_detail_pend_reason_jobnum_daily.xml
	IsfDetailPendingReasonTimeDail y	datatransformer/flow/lsfdetailpendreason/daily/hostreason/ pendtime/main_detail_pend_reason_pendtime_daily.xml
	hardwareCoreUtilDaily	datatransformer/flow/hardware_with_core_util-sp006/daily/ main_hardware_core_util_daily.xml

Task	Data transformer name	Data flow entry
7	licUsageByUserDaily	datatransformer/flow/flexlmusage/daily/licenseusagedaily/ license_usage_daily_etl.xml
	licUtilDaily	datatransformer/flow/flexlmusage/daily/licenseutildaily/ license_util_daily_etl.xml
	licUtilByFeatureDaily	datatransformer/flow/flexlmusage/daily/licenseutilfeaturedaily/ lic_util_daily_feature_etl.xml
	licUsageByUsergrpDaily	datatransformer/flow/flexlmusage/daily/ licenseusagebyusrgrpdaily/license_usage_daily_byusrgrp_etl.xml
	licConsumpDaily	datatransformer/flow/flexlmusage/daily/licenseconsumpdaily/ license_consumption_daily_etl.xml
	licDenialDaily	datatransformer/flow/flexlmevents/daily/liceventsdaily/ main_license_denial_daily.xml
	licDenialByFeatureDaily	datatransformer/flow/flexlmevents/daily/liceventsdaily/ main_license_denial_by_feature_daily.xml
	flexEventDaily	datatransformer/flow/flexnetevents/daily/flexeventdaily/ main_flexEventDaily.xml
	IsfFlexEventDaily	datatransformer/flow/flexnetevents/daily/Isfflexeventdaily/ main_IsfFlexEventDaily.xml
	licDenialDaily	datatransformer/flow/flexnetevents/daily/flexeventdenialdaily/ main_license_denial_daily.xml
	licDenialByFeatureDaily	datatransformer/flow/flexnetevents/daily/flexeventdenialdaily/ main_license_denial_by_feature_daily.xml

1. Log into the Analytics server host.
2. Create and enable a new scheduled task in the Analytics Console.
 - a) Launch the Analytics Console.
 - UNIX: **ANALYTICS_TOP/bin/runconsole.sh**
 - Windows: Start > Programs > Platform Analytics Server > Platform Analytics Console
 - b) Click Scheduled Tasks in the navigation tree.
 - c) Right-click on the main window and select Add Scheduled Task.
 - d) Complete the required fields for the new task.
 - Scheduled Task: Specify the name of this task.
 - Script File: Specify **bin/dataagghourly.js** for hourly tasks or **bin/dataaggdaily.js** for daily tasks.
 - Script Function: Specify **doit**.

For example, if you are creating Task 2 from the table with the recommended format of splitting data transformers, specify the following:

- Scheduled Task: Specify Task2 as the name of the scheduled task.
- Script File: Specify **bin/dataagghourly.js** as the path to the script file.
- Script Function: Specify **doit** as the script function.

- e) Enable the new scheduled task that you created.
3. In the `tasks` subdirectory of `ANALYTICS_TOP`, create a new directory with the same name as the name of the new scheduled task and navigate to the new directory.

For example, for Task 2 on a UNIX host,

```
cd ANALYTICS_TOP/tasks
```

```
mkdir Task2
```

```
cd Task2
```

4. From the new directory, create a text file of any name with the `.tsk` extension.

For example, create `task2.tsk`.

5. In the new `.tsk` text file, for each data transformer that you would like the scheduled task to control, add its corresponding data flow entry as a new file to the line.

You can also add a comment with the name of the data transformer if you start the line with the `#` character.

For example, for Task 2, the `task2.tsk` file should contain the following lines:

```
# IsfJobSlotsUsageByGroupHourly  
datatransformer/flow/bjobs-sp012/hourly/jobdetailresusage/main_job_slotsusage_bygroup.xml  
# IsfJobStatisticsHourly  
datatransformer/flow/bjobs-sp012/hourly/jobstatistics/main_jobstatistics_etl.xml
```

Optimize the data purger (non-partitioned data schema only)

To optimize the data purger to the recommended level, you must be using a cluster with less than 1000 hosts and have approximately 40 GB of UNDO tablespace in the database.

If you are using a non-partitioned data schema in a cluster with less than 1000 hosts, you can change the data purger batch size to improve the performance of the data purger. If you have a cluster of more than 1000 hosts, you should use partitioned data schema.

If you are using a partitioned data schema, your data purger is already operating at optimal levels, and the following changes will not improve its performance.

1. Log into the Analytics server host.
2. Edit the `pi.conf` file.
 - UNIX: `ANALYTICS_TOP/conf/pi.conf`
 - Windows: `ANALYTICS_TOP\conf\pi.conf`
3. Change the following parameter definition to the recommended level:

```
PURGER_BATCH_SIZE=1000000
```

To set the parameter to this level, you must be using a cluster with less than 1000 hosts and have approximately 40 GB of UNDO tablespace in the database.

If your cluster is greater than 1000 hosts, you should use a partitioned data schema, as the data purger cannot function at optimal performance with non-partitioned data schema for a cluster of this size.

Change the data retention period

A long data retention period can have a significant impact on data transformer and cube build performance as well as the data volume. You can tailor the data retention period according to your business requirements to maximize the performance of your Analytics server.

The steps you take to change the data retention period depend on whether you are using partitioning for the data schema.

- [Change the data retention period for non-partitioned data schema](#) on page 40
- [Change the data retention period for partitioned data schema](#) on page 41

Change the data retention period for non-partitioned data schema

If you are using non-partitioned data schema, the data purger consists of two scheduled tasks (PurgeRawData and PurgeAggData), which are disabled by default. To purge out-of-date data, enable these two scheduled tasks using the Analytics Console.

The following table describes the default data retention period grouped by data type. Each configuration file is located in the purger configuration directory:

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

Data type	Default data retention period (days)	Configuration files
Raw data	14	<ul style="list-style-type: none"> • <code>bjobs-sp012.xml</code> • <code>purger_hardware_with_core_util-sp006_rawdata.xml</code> • <code>purger_license_rawdata.xml</code> • <code>purger_lsf_advanced_rawdata.xml</code> • <code>purger_lsf_basic_rawdata.xml</code>
Job- and license-related hourly data	84	<ul style="list-style-type: none"> • <code>purger_hardware_with_core_util-sp006_hourlydata.xml</code> • <code>purger_histogram-sp011.xml</code> • <code>purger_license_hourlydata.xml</code> • <code>purger_lsf_job_hourlydata.xml</code>
Host-related hourly data	14	<ul style="list-style-type: none"> • <code>purger_lsf_host_hourlydata.xml</code>
Daily data	182	<ul style="list-style-type: none"> • <code>purger_hardware_with_core_util-sp006_dailydata.xml</code> • <code>purger_license_dailydata.xml</code> • <code>purger_lsf_dailydata.xml</code>

1. Log into the Analytics server host.
2. Enable the PurgeRawData and PurgeAggData scheduled tasks using the Analytics Console.
 - a) Launch the Analytics Console.
 - UNIX: `ANALYTICS_TOP/bin/runconsole.sh`

- Windows: Start > Programs > Platform Analytics Server > Platform Analytics Console
- b) Click Scheduled Tasks in the navigation tree.
 - c) Enable the PurgeRawData and PurgeAggData scheduled tasks.
3. Edit each configuration file for the data type that you want to change.

Within each configuration file, navigate to the TableList element and change the Duration attribute to the duration that you want.

For example, to change the data retention period to 28 days, change the Duration attribute to the following.

```
<TableList ... Duration="28" ... />
```

Change the data retention period for partitioned data schema

If you are using partitioned data schema, the data purger consists of two scheduled tasks (RawTablePartition and WideTablePartition), which are enabled by default.

The following table describes the default data retention period grouped by data type. Each configuration file is located in the bin directory:

- UNIX: *ANALYTICS_TOP/bin*
- Windows: *ANALYTICS_TOP\bin*

Data type	Default data retention period (days)	Configuration file	Command to purge the data
Host-related raw data	5	autopartition_raw.js	PA_PARTITION.DROP_PARTITION('RAW_HOST', 1, SYSDATE + 2, 7, 1)
Job- and license-related raw data	14	autopartition_raw.js	PA_PARTITION.DROP_PARTITION('RAW', 1, SYSDATE + 2, 16, 1)
Host-related hourly data	18	autopartition_wi.js	PA_PARTITION.DROP_PARTITION('HOURLY_HOST', 2, SYSDATE + 2, 10, 2)
Job- and license-related hourly data	98	autopartition_wi.js	PA_PARTITION.DROP_PARTITION('HOURLY', 2, SYSDATE + 2, 14, 7)
Daily data	222	autopartition_wi.js	PA_PARTITION.DROP_PARTITION('DAILY', 2, SYSDATE + 2, 8, 28)

In the commands to purge the data, the fourth parameter specifies the number of partitions for each table to be kept, and the fifth parameter specifies the number of days of data that is kept; the data purger deletes all remaining older data. Therefore, for host-related raw data, the default is 16 partitions kept and the data retention period is one day.

To change the default retention period for each data type, edit the configuration file and navigate to its corresponding command to purge the data. Change the fifth parameter to the new retention period.

Optimize the Analytics server

Index

C

- customizations 4
 - node 5
 - server 5
- customizations
 - database schema 4

D

- data loaders
 - log files 21
 - verifying 21
- database
 - schema
 - customizations 4
- display environment
 - setting 14

E

- environment
 - display
 - setting 14
 - PERF
 - sourcing 20

F

- files

- installation
 - Analytics 8

L

- loader controller
 - log file 21

O

- Oracle
 - database
 - schema 8

P

- PERF
 - services
 - starting 20
 - verifying 20

S

- scheduled tasks 17, 18
- services
 - starting 20
 - verifying 20
- source environment
 - PERF 20