

---

# Installing Platform Analytics

Platform Analytics  
Version 8.0  
Release date: May 2011



Copyright

© 1994-2011 Platform Computing Corporation.

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

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

We'd like to hear from you

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

Your comments should pertain only to Platform documentation. For product support, contact [support@platform.com](mailto:support@platform.com).

Document redistribution and translation

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

Internal redistribution

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

Trademarks

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

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

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

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

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

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

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

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

Third-party license agreements

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

---

# Contents

1	Platform Analytics hosts .....	5
	System ports .....	6
	System architecture diagrams .....	8
2	Licensing .....	11
3	Prepare for the Platform Analytics installation .....	13
	What you need to do .....	14
4	Install the database .....	19
	What you need to do .....	20
5	Install the Platform Analytics server .....	23
	What you need to do .....	24
6	Install the Platform Analytics node .....	27
	What you need to do .....	28
7	Install the Platform Analytics reporting server .....	31
	What you need to do .....	32
	Setup your cluster locations .....	33
8	Prepare the Platform Application Center host .....	35
	What you need to do .....	36
9	Verify and configure the Platform Analytics server .....	37
	What you need to do .....	38
10	Configure the Platform Analytics node .....	39
	What you need to do .....	40
11	Optimize the Platform Analytics server .....	47
	What you need to do .....	48
12	Optimize the Platform Analytics node .....	51
	What you need to do .....	52
13	Remove Platform Analytics software from the host .....	57
	What you need to do .....	58



## Platform Analytics hosts

There are three types of Platform Analytics hosts that you install:

Platform Analytics server	The Platform Analytics server manages the data that the Platform Analytics nodes collect. You can perform all server functions using the Platform Analytics Console in the server.
Platform Analytics reporting server	The Platform Analytics reporting server generates reports based on the data that the Platform Analytics nodes collect. Users can view these reports using a web browser. The Platform Analytics reporting server can run on the same host as the Platform Analytics server if this host meets the system requirements of both the Tableau Server and the Platform Analytics server software.
Platform Analytics node	Platform 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 (including license data if connected to a license server), or is a standalone host that collects license data.

## System ports

Platform Analytics hosts and components use the following system ports. You need to ensure that the system ports for Platform Analytics are open for the hosts and components to communicate with one another. You also need to ensure that your web server port is open.

### Platform Analytics system ports

Platform Analytics hosts use the following TCP ports by default. The following tables describe the configuration files that you need to modify to change the default settings and the firewall configurations you need to set to allow communication using these ports between the various hosts.

#### Platform Analytics server ports

Port name	Default port number	Description	Firewall configuration (all bi-directional)	Configuration file
PIAM_PORT	9091	Internal port for the task scheduler.	N/A	<i>ANALYTICS_TOP</i> /conf/pi.conf
PIEM_PORT	9092	The Platform Analytics event manager uses this port to receive events from the Platform Analytics server and nodes.	Allow connections for this port from each node host to the server host.	<i>ANALYTICS_TOP</i> /conf/pi.conf
Remoting server port	9093	The Platform Analytics remoting server uses this port to receive data from the remoting node.	Allow connections for this port from the remoting node to the server host.	N/A

#### Platform Analytics node ports

Port name	Default port number	Description	Firewall configuration (all bi-directional)	Configuration file
PLC port	4046	The Platform Analytics server uses this port to manage the loader controller (plc) on the Platform Analytics node.	Allow connections for this port from the server host to each node host.	<i>ANALYTICS_TOP</i> /conf/plc.xml

#### Third-party tool ports

Port name	Default port number	Description	Firewall configuration (all bi-directional)
Web server port	80	End users can use this port to browse reports.	Allow connections for this port from each browsing host to the server host.
License server port	N/A	To collect license usage data from the FLEXnet server, the Platform Analytics node needs to communicate with the FLEXnet server host via TCP.	Allow connections for this port from the node host to each FLEXnet server host.
FLEXnet Manager database port	1433 for MS SQL Server	To collect data from FLEXnet Manager, the Platform Analytics node needs to communicate with the FLEXnet Manager database via TCP.	Allow connections for this port from the node host to the FLEXnet Manager database host.

# System architecture diagrams

The following diagrams show architectures of Platform Analytics systems that monitor multiple LSF clusters.

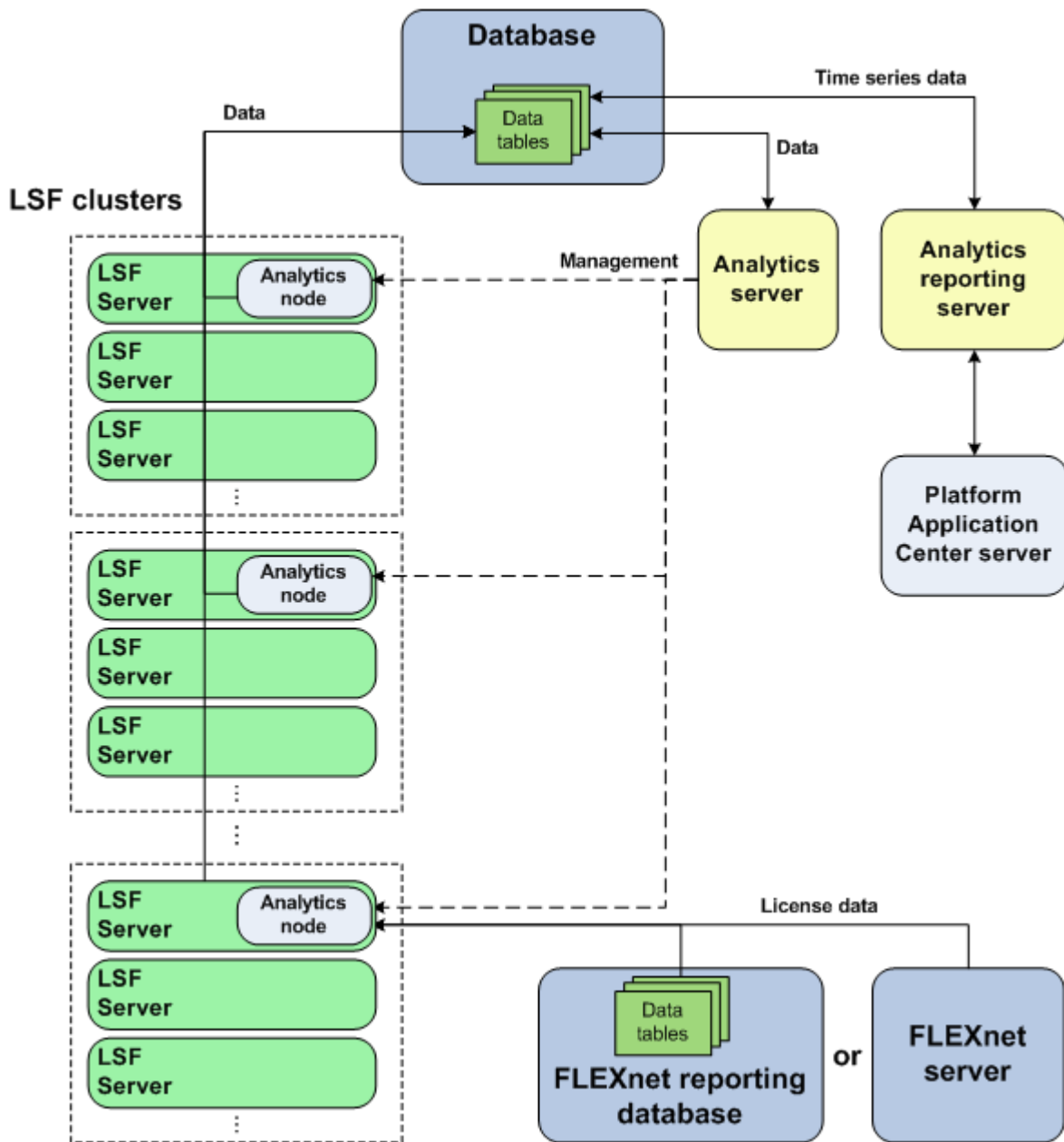


Figure 1: Platform Analytics system architecture where each node is a member of an LSF cluster



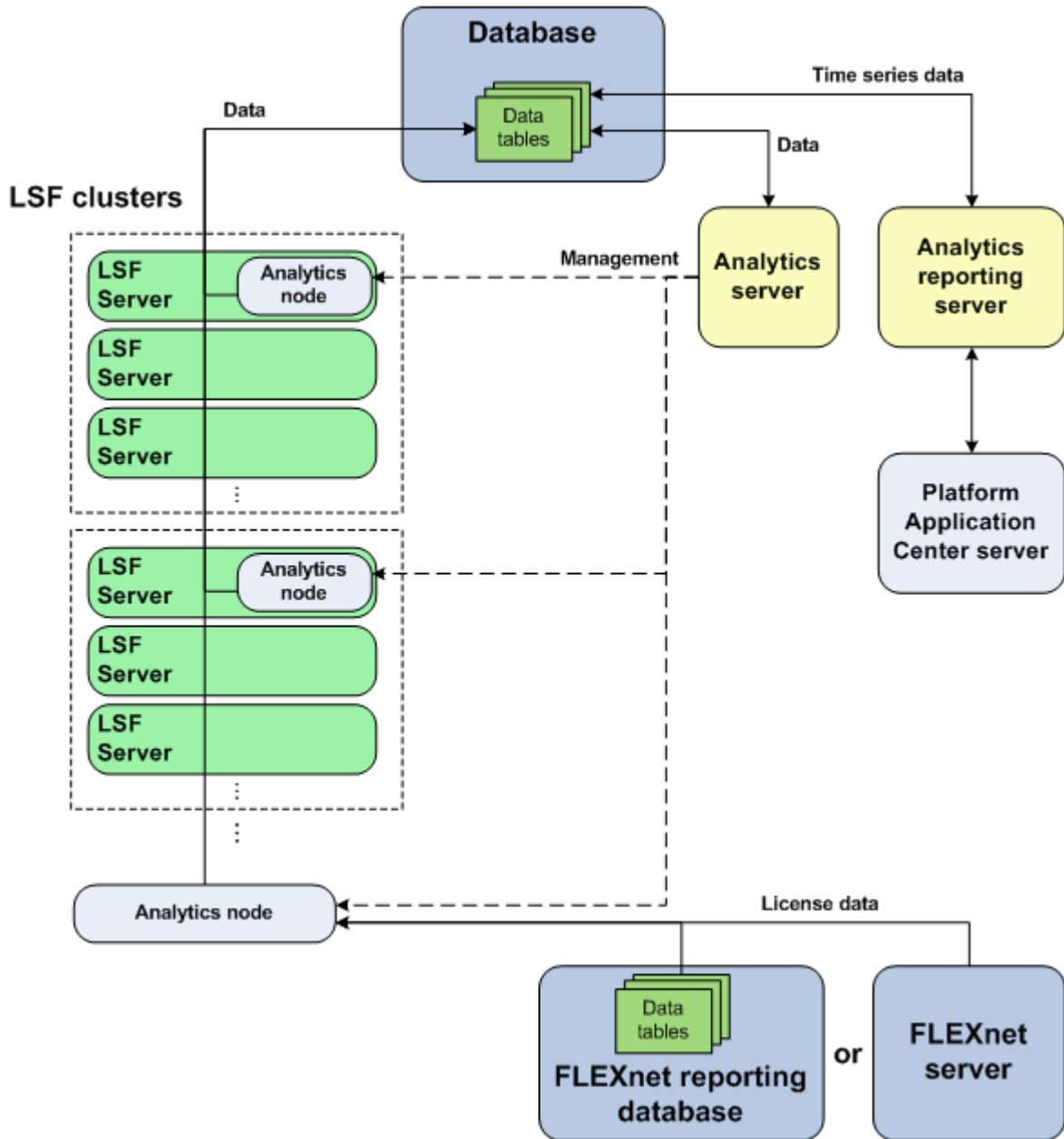


Figure 2: Platform Analytics system architecture with a standalone node connected to a license server

Platform Analytics hosts

## Licensing

The Platform Analytics license file includes licenses for data collection.

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

Analytics base	This is a must have license for Platform Analytics. This license allows you to collect data from LSF clusters.
LSF advanced data collection	This license allows you to collect LSF advanced data from LSF clusters. LSF advanced data is cluster performance and operation data that is not gathered in the base PERF package included with LSF.
License data collection	This license allows you to collect license usage and event data from your FLEXnet servers.
Vertica database connector	This license allows you to control Analytics database connector.

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



## Prepare for the Platform Analytics installation

Select the hosts that meet the detailed system requirements, prepare the Platform LSF cluster, and download the installation packages for Platform Analytics.

## What you need to do

1. [Select the database hosts](#) on page 14
2. [Select the Platform Analytics server host](#) on page 15
3. [Select the Platform Analytics reporting server host](#) on page 16
4. [Select the Platform Analytics node hosts](#) on page 17
5. [Prepare the Platform LSF clusters \(Platform LSF 7.0.4, 7.0.5, 7.0.6, and 8.0 only\)](#) on page 17
6. [Obtain the necessary installation files](#) on page 18

## Select the database hosts

For optimal performance of your production database, the Vertica database cluster should consist of at least three dedicated multi-core hosts running on a high-bandwidth network. Since the Vertica database needs to share a large volume of data among the database nodes in the database cluster during data loading or data querying, network bandwidth is an important performance bottleneck for a production database. Therefore, the Vertica database cluster should have a Gigabit Ethernet connection with the Platform Analytics reporting server and the Platform Analytics node hosts.

Vertica recommends a 1-10 GB full duplex switch for the private network interface and a VLAN or separate switch for the public network. The switch used for the private network should have sufficient bandwidth to enable 1 GB transfer speeds between any pair of nodes.

The hardware requirements are the same for all the intended database hosts. Refer to the Vertica documentation, or the *Release Notes for Platform Platform Analytics* for the latest list of system requirements and supported operating systems for the Vertica database hosts.

Input/output (I/O) performance is important for the operation of a database while fault tolerance is important to safeguard your data. Using a RAID 01 or 10 system enables the database host to take advantage of data striping and data mirroring. Data striping allows data to be transferred to multiple hard disks concurrently, which improves input/output (I/O) performance. Data mirroring means that your database does not lose data even if a hard disk fails.

The following table describes the optimal configuration of the database depending on the size of your cluster. The specific hardware recommendations for each database host are the same:

Cluster size	Number of hosts	RAM	CPU	Local hard disk	Network
Medium (100 - 1000 hosts)	3	16 GB	4 x 2.4GHz	10000 RPM SATA/SCSI/SAS/ SSD RAID 01 or 10 300 GB	Gigabit Ethernet
Large (more than 1000 hosts)	more than 3	32 GB	8 x 2.4GHz	10000 RPM SATA/SCSI/SAS/SSD RAID 01 or 10 1 TB	Gigabit Ethernet

## Data striping

Data striping is the technique of segmenting logically-sequential data, such as a single file, so that the database can assign segments to multiple physical devices (usually disk drives for RAID storage, or

network interfaces for grid-oriented storage) in a round-robin fashion and thus be read or written concurrently.

Automatic data striping is available in certain RAID devices under software or hardware control, and in file systems of clusters. The following parameters are important when improving I/O performance:

#### Stripe width

The number of parallel stripes that can be written to or read from simultaneously. This is the number of disks in the RAID system, and as it increases, the read/write performance of striped data also increases.

#### Stripe size

The size of the stripes written to each disk. This may also be referred to as block size, chunk size, stripe length, or granularity.

You should use a large stripe size of at least 1 MB.

If you are using RAID devices, you should use RAID 10 or RAID 01, because it offers the best performance of all RAID systems and good fault tolerance.

## Select the Platform Analytics server host

When selecting a host to be the Platform Analytics server, you need to ensure that the host is running a supported operating system. Refer to the *Release Notes for Platform Analytics* for the latest list of supported operating systems for the Platform Analytics server host.

#### Tip:

If you select a host that also meets the Tableau Server system requirements, you can also select the Platform Analytics server host to be the Platform Analytics reporting server host.

For optimal performance, the Platform Analytics server host should be a dedicated multi-core host with sufficient memory and input/output performance. The network bandwidth between the Platform Analytics server, the database hosts, and the Platform Analytics nodes is a key performance factor in the Platform Analytics server host.

If you are not using asynchronous data loading mode, the following hardware configuration should be sufficient:

RAM	CPU	Local hard disk	Network
4 GB	4 × 2.4 GHz	7200 RPM SATA/SCSI/SAS 50 GB	Gigabit Ethernet

If you are using asynchronous data loading mode, memory is a key performance factor for the Platform Analytics server host. If the Platform Analytics server is running on a Windows host, you should use the 64-bit version of Windows because Java cannot use more than 1638MB of memory on 32-bit platforms.

You should only use the asynchronous data loading mode for sending data from the Platform Analytics node to the database over a slow or unstable network.

The following table describes the optimal hardware configuration of the Platform Analytics server if you are using asynchronous data loading, depending on the size of your cluster:

Cluster size	RAM	CPU	Local hard disk	Network
Medium (100 - 1000 hosts)	4 GB	4 × 2.4GHz	7200 RPM SATA/SCSI/SAS 50 GB	Gigabit Ethernet
Large (more than 1000 hosts)	8 GB	4 × 2.4GHz	7200 RPM SATA/SCSI/SAS 50 GB	Gigabit Ethernet

**Note:**

Using an NFS disk mount instead of a local hard disk is not recommended.

## Select the Platform Analytics reporting server host

When selecting a host to be the Platform Analytics reporting server host, you need to ensure that the host meets the detailed system requirements for Tableau Server:

Operating system	RAM	CPU	Services	User accounts
Windows Server 2003 (SP2 or higher)	2 GB	Dual-core	Do not run Internet Information Services (IIS) to avoid conflicts with the web server port 80.	Access to an administrator account to install software and services.
Windows Server 2008				Access to a user account that the service can use (optional).
Windows Server 2008 R2				

**Tip:**

If the Platform Analytics server host also meets the Tableau Server system requirements, you can select the Platform Analytics server host to also be the Platform Analytics reporting server host.

Refer to the Tableau Server documentation, or to the *Release Notes for Platform Analytics* for the latest list of system requirements and supported operating systems for the Tableau Server.

The network bandwidth between the Platform Analytics reporting host and the database cluster may be an important performance bottleneck. Therefore, the Platform Analytics reporting host should have a Gigabit Ethernet connection with database hosts.

The following table describes the optimal hardware configuration of the Platform Analytics reporting server, depending on the size of your cluster:

Cluster size	RAM	CPU	Local hard disk	Network
Medium (100 - 1000 hosts)	4 GB	4 × 2.4GHz	7200 RPM SATA/SCSI/SAS 50 GB	Gigabit Ethernet



Cluster size	RAM	CPU	Local hard disk	Network
Large (more than 1000 hosts)	8 GB	4 × 2.4GHz	7200 RPM SATA/SCSI/SAS 50 GB	Gigabit Ethernet

## Select the Platform Analytics node hosts

When selecting a host in the LSF clusters to be an Platform Analytics node, you need to ensure that the host is running a supported operating system, and that it meets the minimum hardware requirements. Refer to the *Release Notes for Platform Platform Analytics* for the latest system requirements for the Platform Analytics node host.

For optimal performance of your Platform Analytics node, the host should be running on a high-bandwidth network. Since network bandwidth is an important performance bottleneck for the Platform Analytics nodes, the Platform Analytics node host should have a Gigabit Ethernet connection with the database host. If the Platform Analytics node is running on a Windows host, you should use the 64-bit version of Windows because Java cannot use more than 1638MB of memory on 32-bit platforms.

The following table describes the optimal hardware configuration of the Platform Analytics node depending on the size of the clusters in which the node resides:

Cluster size	RAM	CPU	Local hard disk	Network
Medium (100 - 1000 hosts)	4 GB	2 × 2.4GHz	7200 RPM SATA/SCSI/SAS 50 GB	Gigabit Ethernet
Large (more than 1000 hosts)	8 GB	4 × 2.4GHz	7200 RPM SATA/SCSI/SAS 50 GB	Gigabit Ethernet

## Prepare the Platform LSF clusters (Platform LSF 7.0.4, 7.0.5, 7.0.6, and 8.0 only)

Your clusters must be running one of the following:

- Platform LSF 7.0.4
- Platform LSF 7.0.5
- Platform LSF 7.0.6
- Platform LSF 8.0

You can skip this step for any clusters that are not running any of these versions of Platform LSF.

By default, Platform LSF 7.0.4, 7.0.5, 7.0.6, and 8.0 do not enable the `lsb.stream` file for the exporting of LSF job event data.

If you want the Platform Analytics node to collect LSF cluster data from your Platform LSF 7.0.4, 7.0.5, 7.0.6, or 8.0 cluster, you need to enable the `lsb.stream` file because Platform Analytics requires this file for the data loaders to obtain job data.

1. Log into a host in the Platform LSF cluster.
2. Edit the `lsb.params` file.
  - UNIX: `$LSF_ENVDIR/lsbatch/cluster_name/configdir/lsb.params`
  - Windows: `%LSF_ENVDIR%\lsbatch\cluster_name\configdir\lsb.params`
3. In the `lsb.params` file, edit the Parameters section to enable the exporting of LSF job event data to the `lsb.stream` file.

Add the following lines to the Parameters section:

```
# Enable streaming of lsbatch system events
ENABLE_EVENT_STREAM=y
# Determines the location of the lsb.stream file. This parameter is optional.
# The default location is: $LSB_SHAREDIR/{clustername}/logdir/stream.
# EVENT_STREAM_FILE=/tmp/lsb.mystream
# Determines the maximum size of the lsb.stream file. This parameter is optional.
# The default size is 100MB.
# MAX_EVENT_STREAM_SIZE=10000
```

4. Reconfigure `mbatchd` to apply these changes.  
**badmin mbdrestart**
5. To verify that these changes are in effect, verify that the `lsb.stream` files exists.

By default, `lsb.stream` is located at the following directories:

- UNIX: `$LSB_SHAREDIR/cluster_name/logdir/stream`
- Windows: `%LSB_SHAREDIR%\cluster_name\logdir\stream`

If you defined the `EVENT_STREAM_FILE` parameter in `lsb.params`, check the specified file path for the `lsb.stream` file.

## Obtain the necessary installation files

1. Obtain the necessary files for installing Platform Platform Analytics.

You need the following files to install Platform Platform Analytics:

- Platform Platform Analytics server installation package
- Platform Platform Analytics node installation package
- Platform Platform Analytics documentation package
- Platform Platform Analytics license file
- Platform

2. Obtain the necessary files for installing the Vertica Analytic Database.

You need the following files to install the Vertica Analytic Database:

- Vertica Analytic Database installation package
- Platform Platform Analytics database schema package

3. Obtain the necessary files for installing the Tableau Server.

You need the following files to install the Tableau Server:

- Tableau Server installation package
- Platform Platform Analytics report package

4. Obtain the necessary file for integrating Platform Analytics into Platform Application Center.

README for Integrating Platform Analytics 8.0 into Platform Application Center 8.0.

## Install the database

The Vertica database, which consists of one or more database hosts, contains the cluster operations data for reporting and analysis. You must install and configure the Vertica database properly before you install Platform Analytics.

You can install the Vertica database on a single host or on a cluster of multiple hosts. Installing Vertica on multiple hosts improves the performance and reliability of your database.

## What you need to do

The intended database host (or hosts in the cluster) must meet the Platform Analytics and Vertica system requirements. See [Prepare for the Platform Analytics installation](#) on page 13.

1. Install Vertica onto the intended database host (or hosts in the cluster).

For more details on installing Vertica, refer to the *Installation and Configuration Guide* for the Vertica Analytic Database. Follow the steps described in the *Before You Install* and *Installing Vertica* chapters.

---

**Note:**

You can improve the performance and reliability of your database by installing Vertica on multiple hosts. For best results, Vertica recommends that disk utilization per host should not be more than sixty percent (60%) for a K-Safe=1 database.

2. Change the `create_pa8.0_schema.sql` manually in case that the K-Safe level of your Vertica database is not 1. K-Safe level of your Vertica database is decided by the number of Vertica database nodes:

Number of Vertica database nodes <3: K-Safe = 0

3 ≤ number of Vertica database nodes <5: K-Safe = 1

Number of Vertica database nodes >5: K-Safe = 2

Based on the above settings, check the actual K-Safe level of your vertica database and set it accordingly.

3. Log into a database host as the Vertica database administrator.
4. Create a new user and grant schema creation privileges.
  - a) Create a new user.

- Logon to `vsq1` as the database administrator.

```
/opt/vertica/bin/vsql -w password
```

Where *password* is the database administrator password.

- **create user *username* identified by '*password*';**

Where *username* is the name of the user with which to connect to the database, instead of the default user (the database administrator) and *password* is the password assigning it to the new user.

- b) Grant the schema creation privileges to the created user.

```
grant create on database database_name to username
```

Where *database\_name* is the name of the database and *username* is the name of the created user.

- c) Create a schema as a new user.

- Logon to `vsq1` as the new user

```
/opt/vertica/bin/vsql -U username -w password
```

Where *username* and *password* are new user credentials.

- **create schema *username***

Where *username* should be the same username created for the new user.

5. Extract the database schema package as the database administrator.
6. Launch the `vsq1` command line.
  - a) Navigate to the `bin` subdirectory of the Vertica installation directory.

By default, this is `/opt/vertica/bin`.

- b) Run `vsq1` to connect to the database.

```
.vsq1 -d database_name -p port -U username -w password
```

where

- *database\_name* is the name of the database
- *port* is the TCP port number or the local socket file extension in which the server is listening for connections. The default is port number 5433.
- *username* is the name of the user with which to connect to the database, instead of the default user (the database administrator).
- *password* is the password for the database user.

Alternately, you can run `vsq1` with no options to accept the defaults and specify the administrator password at the prompt.

7. Create the Vertica database schema.

From the `vsq1` command line, run the `create_pa8.0_schema.sql` script to create a database schema.

```
vi file_path/create_pa8.0_schema.sql
```

where *file\_path* is the file path to the `create_pa_schema.sql` file (schema/Vertica from the directory where you extracted the database schema package).

8. Increase the maximum number of client sessions.

You need to increase the maximum number of client sessions for the database to communicate with the Platform Analytics nodes. In Vertica, this is the `SESSIONS` parameter, which you can change from the `vsq1` command line by using the `SET_CONFIG_PARAMETER` function:

```
SELECT SET_CONFIG_PARAMETER('MaxClientSessions', SESSIONS_value);
```

Each Platform Analytics node needs approximately 30 sessions and the Platform Analytics server needs approximately 10 sessions. In addition, each intended user that will be concurrently accessing the live Platform Analytics reports will also need a client session.

Therefore, for optimal performance of the database, calculate the optimal value of the `SESSIONS` parameter in Vertica as follows:

$$\text{SESSIONS\_value} = (\# \text{ of Platform Analytics nodes}) \times 30 + 10 + (\# \text{ of concurrent users browsing live reports})$$

For example, if you have three Platform Analytics nodes, and will have an extra ten users concurrently access live Platform Analytics reports, increasing the maximum number of client sessions to 110 should be sufficient. From the `vsq1` command line, run the following command:

```
SELECT SET_CONFIG_PARAMETER('MaxClientSessions', 110);
```

9. Modify the default queue timeout of the `tm` resource pool to 90 seconds.

From the `vsq1` command line, run the following command:

```
ALTER RESOURCE POOL tm queue_timeout 90;
```

10. To obtain optimal performance, store TEMP and DATA in different storage locations. Temp data is distributed across available storage locations based on available storage space. However, data can be stored on different storage locations based on predicted or measured access patterns.
11. For each database host in the cluster, set the blockdev size.

The block device (blockdev) is the physical storage device on the database. Set the blockdev size to obtain optimal performance by running the following commands for the drives in which your data directory is located:

```
sudo blockdev --getss drive
```

```
sudo blockdev --getra drive
```

For example,

```
sudo blockdev --getss /dev/md0
```

```
sudo blockdev --getra /dev/md0
```

By default, the first command should return 512. The second command should return 2048, which means the readahead parameter is set to 1 MB.

If the readahead parameter is set too high, the database host may experience a slow mergeout.

12. Optional. If the readahead parameter is set too high, set the blockdev size to correct this problem.

```
sudo blockdev --setra 2048
```

To retain this setting every time the host restarts, copy this line into the `/etc/rc.local` file.

## Install the Platform Analytics server

Install the Platform Analytics server on a UNIX or Windows host.

## What you need to do

Check the following:

- The Platform Analytics server host meets the detailed system and software requirements.  
For more details, refer to [Prepare for the Platform Analytics installation](#) on page 13.
- The Platform Analytics server host operating system can handle daylight savings time correctly.
- You removed any previous installations of Platform Analytics from your host.
- If you are installing on a UNIX host, your host must have access to the X-Windows environment:
  - If you are connected to the intended Platform Analytics server host locally, the host must be running X-Windows.
  - If you are connected to the intended Platform Analytics server host via `telnet`, you must be running `xserver` on your local host.
- The required ports for the Platform Analytics and database hosts are free. If a firewall exists, all these ports must be open in the firewall. All the required ports are TCP.

For more details, refer to [Platform Analytics hosts](#) on page 5.

- The Platform Analytics server host has access to the Platform Analytics license file (`license.dat`).
1. Log into the intended Platform Analytics server host.
    - In UNIX, if you log in as root, you will be able to update the `rc` boot scripts and specify the user name and group of the Platform Analytics administrator during the installation.  
Otherwise, log in as the intended Platform Analytics administrator account.
    - In Windows, log in as the intended Platform Analytics administrator account.
  2. If you connected to the UNIX host via `telnet` 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 `csch` 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 Platform Analytics server installation package.
  - UNIX: **`./platform_analytics8.0_server_platform.bin`**
  - Windows: **`platform_analytics8.0_server_platform.exe`**
4. Follow the dialog prompts to complete the installation.

To install the Platform Analytics server, you may need to input the following:

- Installation directory  
Accept the default file path or specify your own installation directory.



In this documentation, *ANALYTICS\_TOP* refers to the top-level Platform Analytics installation directory in your host.

- Base communication ports

The Platform Analytics server requires the use of two consecutive ports, starting with the base port. The default base port is 9091, therefore, the Platform Analytics server uses ports 9091 and 9092 by default.

- Email notification settings

You can receive notifications of events via email. This means you will be made aware of potential problems in your cluster without having to constantly monitor the Platform Analytics Console.

After enabling email notifications, specify the SMTP mail server and the intended recipient of the email notifications. The intended sender is the account that is used to send the email notifications.

---

**Note:**

You should enable the email notification during the initial installation rather than at a later time. If you do not enable email notification, you will need to check the Platform Analytics Console regularly to make sure that Platform Analytics is working properly.

---

- Database connection properties

The user name and password is generally the account that you used to create your database schema.

- Set RC properties, if you installed on a UNIX host as root.

Enable this setting to update the `rc.boot` script.

5. If you are installing on a UNIX (Solaris or Linux) host, start the Platform Analytics server.

**ANALYTICS\_TOP/bin/perfadmin start all**

6. Verify that the `pat.s` service has started.

- UNIX: **ANALYTICS\_TOP/bin/perfadmin list**
- Windows: **ANALYTICS\_TOP\bin\perfadmin list**

Install the Platform Analytics server

## Install the Platform Analytics node

Install the Platform Analytics node package on a UNIX or Windows host. You need to have one Platform Analytics node host in each cluster that Platform Analytics will handle.

## What you need to do

Check the following:

- The Platform Analytics node host meets the detailed system and software requirements.  
For more details, refer to [Prepare for the Platform Analytics installation](#) on page 13.
- The Platform Analytics node host operating system can handle daylight savings time correctly.
- If you want the Platform Analytics node to collect LSF cluster data, check the following:
  - You have access to the primary LSF administrator account.
  - The Platform Analytics node host must be an LSF server in the cluster with access to the LSF event file (`lsb.stream`).
- If you are installing on a UNIX host, your host must have access to the X-Windows environment:
  - If you are connected to the intended Platform Analytics node host locally, the host must be running X-Windows.
  - If you are connected to the intended Platform Analytics node host via `telnet`, you must be running `xserver` on your local host.
- The required port for the Platform Analytics hosts is free. If a firewall exists, this TCP port must be open in the firewall.

For more details, refer to [Platform Analytics hosts](#) on page 5.

- The Vertica database host is properly configured and running.
- The Platform Analytics node host has access to the Vertica JDBC driver.
- The Platform Analytics node host has access to the Platform Analytics license file (`license.dat`).

### 1. Log into the intended Platform Analytics node host.

- In UNIX, if you want to update the `rc.boot` scripts, log in as `root`.
- In Windows, log in as the intended Platform Analytics administrator account. If you want the Platform Analytics node to collect LSF cluster data, you must log in as the primary LSF administrator.

### 2. If you connected to the UNIX host via `telnet` 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. If you are installing on a UNIX host and you want the Platform Analytics node to collect LSF cluster data, source the LSF environment.

- For `csh` or `tcsh`: **source `LSF_TOP/conf/cshrc.lsf`**

- For sh, ksh, or bash: **. LSF\_TOP/conf/profile.lsf**
4. Run the Platform Analytics node installation package.
    - UNIX: **./platform\_analytics8.0\_node\_platform.bin**
    - Windows: **platform\_analytics8.0\_node\_platform.exe**
  5. Follow the dialog prompts to complete the installation.

To install the Platform Analytics node, you may need to input the following:

- Platform Analytics license file location
- Data collection settings
- Installation directory

Accept the default file path or specify your own installation directory.

In this documentation, *ANALYTICS\_TOP* refers to the top-level Platform Analytics installation directory in your host.

- The type of data to collect from the cluster
- Directory containing the `lsf.conf` file

By default, this is the value of the `LSF_ENVDIR` environment variable.

- Data source (database host) properties

The user name and password is generally the account that you used to create your data schema.

Replace the sample values in the URL field with the values for your database.

- Loader controller service port

You need to specify a port for the loader controller (`plc`), which uses 4046 by default.

- Set RC properties, if you installed on a UNIX host as `root`.

Enable this setting to update the `rc.boot` script.

6. If you are installing on a UNIX host, source the PERF environment.
  - For `cs`h or `tc`sh: **source ANALYTICS\_TOP/conf/cshrc.perf**
  - For sh, ksh, or bash: **. ANALYTICS\_TOP/conf/profile.perf**
7. If you install Platform Application Center in the same cluster as Platform Analytics, disable the auto delete function of LSF events loader (`lsfeventsl oader`) from Platform Application Center to prevent the `lsb.stream. UTC` files from being deleted automatically.
  - Edit the `$PERF_CONFDIR/conf/dataloader/lsfevents.properties` file of Platform Application Center.
  - Add **AUTO\_DELETE\_STREAM\_FILE=N**
8. Manually stop and start the `plc` service.
 

```
perfadmin stop plc
perfadmin start plc
```
9. Verify that the data loaders are running.
  - Windows: **plcclient -s**
  - UNIX: **plcclient.sh -s**
10. Verify that there are no errors in any of the data loader log files.

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

Install the Platform Analytics node

- UNIX: \$PERF\_LOGDIR/dat al oader
- Windows: %PERF\_LOGDIR%\dat al oader

## Install the Platform Analytics reporting server

Install the Platform Analytics reporting server on a Windows host by installing Tableau Server and deploying the Platform Analytics report package.

## What you need to do

The intended Platform Analytics reporting host must meet the Tableau Server system requirements. For more details, refer to [Prepare for the Platform Analytics installation](#) on page 13.

You can also install the Platform Analytics reporting server on the Platform Analytics server host if the Platform Analytics server host is running Windows.

1. Log into the intended Platform Analytics reporting host as a user with administrator privileges.
2. Run the Tableau Server installation package.

Navigate to the Tableau Server installation package and double-click the file.

3. Follow the dialog prompts to complete the installation.
4. Enable embedded credentials in Tableau Server.
  - a) Log into Tableau Server as the administrator.
  - b) Click Maintenance in the Administration area on the left side of the page.
  - c) Enable the Embedded Credentials option in the Settings section of the Maintenance page.
5. Extract the Platform Analytics report package to a directory in the Platform Analytics reporting server host.

For example, extract the Platform Analytics report package to `C:\analytics8.0_reports`.

6. Navigate to the `conf` subdirectory of the extracted Platform Analytics report package directory.

For example,

```
cd \analytics8.0_reports\conf
```

7. Edit the `rptbuilder.conf` file to enable and specify the value of the required parameters.
8. Publish the Platform Analytics reports to Tableau Server.
  - a) Navigate to the `bin` subdirectory of the extracted Platform Analytics report package directory.

For example,

```
cd \analytics8.0_reports\bin
```

- b) Run the report installer batch file.

```
rptinstall.bat
```
- c) Follow the dialog prompts to complete the installation.

You need to specify the following to complete the installation:

- Database server name.
- Database user name and password.
- Tableau Server administrator name and password.
- Tableau Server port number.

To publish customized workbooks, you have to install and use the Platform Analytics Designer or run the command line provided by the Tableau server.



## Setup your cluster locations

You should have the Platform Analytics Designer installed.

You need to set up all your cluster locations using the Platform Analytics Designer.

1. Open a workbook in the Platform Analytics Designer.
2. Click Presentation Mode on the tool bar.
3. Navigate to the dashboard that shows the map, right-click on it and select Go to Sheet.
4. In Dimensions, select Cluster, right-click on it and select Geographic Role > Edit Locations....
5. In the Edit locations window, select the appropriate location for each data value from the drop-down list.
6. Click OK to save and exit from the window.

Now all the clusters will be shown on the map. If you have many clusters, you can adjust the size of the map to see all your cluster locations.

Install the Platform Analytics reporting server

## Prepare the Platform Application Center host

Install and configure the Platform Application Center host to work with Platform Analytics.

## What you need to do

Platform Application Center allows users and administrators to monitor hosts and to submit and monitor jobs. It also works view the Platform Analytics reports to monitor the Platform LSF clusters.

1. Install Platform Application Center on the intended host.

For more details, refer to the Platform Application Center documentation. The Platform Application Center installation package and documentation files are available from the following URL:

<http://my.platform.com/products/platform-lsf-family/platform-application-center>

If you already have Platform Application Center installed, you can skip this step.

2. Integrate Platform Analytics into Platform Application Center.

For more details, refer to the *README for Integrating Platform Analytics 8.0 into Platform Application Center 8.0*.

## Verify and configure the Platform Analytics server

Use the Platform Analytics Console to verify and configure the Platform Analytics server.

## What you need to do

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

To view the data loader properties, right-click each loader controller instance and select Loader Properties.
4. Click Scheduled Tasks in the navigation tree and modify the times that the scheduled tasks are running, if necessary.
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.
7. If you intend to use the asynchronous data loading mode, enable and configure the remoting server.

You should only use the asynchronous data loading mode for sending data from the Platform Analytics node to the database over a slow or unstable network.

  - a) Navigate to the `ANALYTICS_TOP/conf/` directory.
  - b) Rename the `remotingserver.TMPL` file to `remotingserver.xml`.
  - c) Edit the `remotingserver.xml` file and change the IP address to the IP address of the Platform Analytics server host.
  - d) Edit the `log4j.properties` file and uncomment `RemotingServer` line.
  - e) Rename the `wsm/pars.TMPL` file to `wsm/pars.conf`.
  - f) Start up the remoting server.

**perfadmin start pars**
  - g) If you are on a Windows host and want the remoting server to start up automatically when Windows starts up, change the startup type of the `pars` Window service from Manual to Automatic.

## Configure the Platform Analytics node

Configure the Platform Analytics node to work with your cluster to work together efficiently.

## What you need to do

1. [Stop the redundant PERF services \(in Platform LSF 7 clusters only\)](#) on page 40
2. [Configure the Platform LSF cluster for the host core utilization data loader](#) on page 40
3. [Enable FLEXnet data loaders in your cluster](#) on page 41
4. [Enable FLEXnet Manager data loaders in your cluster](#) on page 42
5. [Install and configure the Platform FLEXnet Manager scripts](#) on page 43
6. [Enable and configure the remoting server client \(asynchronous data loading mode only\)](#) on page 44

## Stop the redundant PERF services (in Platform LSF 7 clusters only)

You can only stop the PERF services if you installed Platform Analytics node in a cluster running Platform LSF 7.

Platform Analytics runs its own PERF services independent of the cluster. Since Platform LSF 7 clusters also have PERF services running, you should stop the redundant PERF services to avoid unnecessary redundancy in your cluster.

1. Log into any host in the LSF cluster as the cluster administrator.
2. If you logged into a UNIX host, source the LSF environment.
  - For `csh` or `tcsh`: **source `LSF_TOP/conf/cshrc.lsf`**
  - For `sh`, `ksh`, or `bash`: **. `LSF_TOP/conf/profile.lsf`**
3. Navigate to the PERF binary directory.
  - UNIX: **cd `$PERF_TOP/version_number/bin`**
  - Windows: **cd `%PERF_TOP%\version_number\bin`**
4. Stop the loader controller (`pl c`) and data purger (`purger`) services.  
**perfadmin stop plc**  
**perfadmin stop purger**
5. Rename the loader controller configuration file (`pl c.xml`) to another file extension to remove it from the list of LSF services.  
For example, rename `pl c.xml` to `pl c.xml.bak`.
6. Rename the data purger configuration file (`purger.xml`) to another file extension to remove it from the list of LSF services.  
For example, rename `purger.xml` to `purger.xml.bak`.
7. Restart EGO on the LSF master host to activate these changes.  
**egosh ego restart `lsf_master_host_name`**

## Configure the Platform LSF cluster for the host core utilization data loader

Configure `el i m` in the Platform LSF cluster to allow the host core utilization data loader to collect data from the cluster.



1. Log into the Platform Analytics node host.
2. If you logged into a UNIX host, source the LSF environment.
  - For `csh` or `tcsh`: **source *LSF\_TOP/conf/cshrc.lsf***
  - For `sh`, `ksh`, or `bash`: **. *LSF\_TOP/conf/profile.lsf***
3. Copy the `elim.coreutil` execution files from the Platform Analytics directory to the relevant Platform LSF directory.

- UNIX:

```
cp ANALYTICS_TOP/elim/os_type/elim.coreutil $LSF_SERVERDIR
```

where ***os\_type*** is `aix`, `hpux`, `linux`, or `solaris`, depending on the specific UNIX operating system.

- Windows (Platform LSF 7.0.x and 8.0 only):

```
copy ANALYTICS_TOP\elim\windows\elim.coreutil.exe LSF_TOP\version\etc
```

4. Edit the `LSF_TOP/conf/lsf.shared` file and add `CORE_UTIL` as a resource.

Add the following text to the Resource section:

```
CORE_UTIL String 300 () (Core Utilization)
```

For example,

```
Begin Resource
RESOURCENAME TYPE INTERVAL INCREASING DESCRIPTION
CORE_UTIL String 300 () (Core Utilization)
End Resource
```

5. Edit the `LSF_TOP/conf/lsf.cluster.cluster_name` file and add `CORE_UTIL` as a resource map.

Add the following text to the Resource section:

```
CORE_UTIL [default]
```

For example,

```
Begin ResourceMap
RESOURCENAME LOCATION
CORE_UTIL [default]
End ResourceMap
```

6. Reconfigure the Platform LSF cluster to apply your changes.

```
badmin reconfig
```

```
lsadmin reconfig
```

## Enable FLEXnet data loaders in your cluster

Check the following:

- You configured Platform Analytics to collect FLEXnet license data.
- The FLEXnet usage data loader and the FLEXnet license server are using the same time zone.

1. Log into the Platform Analytics node host.
2. Modify the FLEXnet usage data loader configuration file to set up a server list pointing to the FLEXnet license servers.
  - a) Edit the FLEXnet usage data loader configuration file.
    - UNIX: `ANALYTICS_TOP/conf/dataloader/flexiusage.properties`

- Windows: `ANALYTICS_TOP\conf\data loader\flexlicusage.properties`
  - b) Change the `ServerList` (or `FileName`) parameter to point to the FLEXnet license servers.
3. Modify the FLEXnet events data loader configuration file to point to the license log file.
    - a) Edit the FLEXnet events data loader configuration file.
      - UNIX: `ANALYTICS_TOP/conf/data loader/flexlicevents.properties`
      - Windows: `ANALYTICS_TOP\conf\data loader\flexlicevents.properties`
    - b) Change the `LicenseLogFile` parameter to point to the FLEXnet license log file. Even for Windows, you should use a slash rather than a backslash.

For example,

- UNIX: `LicenseLogFile=/file_path/lmgrd.log`
  - Windows: `C:/file_path/lmgrd.log`
4. Start the Platform Analytics node host by restarting the loader controller (`plc`) service.

```
perfadmin stop plc
```

```
perfadmin start plc
```

## Enable FLEXnet Manager data loaders in your cluster

You can only enable FLEXnet Manager data loaders if you configured Platform Analytics to collect FLEXnet license data.

1. Log into the Platform Analytics node host.
2. If you connected to the UNIX host via telnet 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 `cs`h or `tc`sh:

```
setenv DISPLAY hostname:0.0
```
- For `sh`, `ksh`, or `bash`:

```
DISPLAY=hostname:0.0  
export DISPLAY
```

where `hostname` is your local host.

3. Add a data source for the FLEXnet Manager Reporting database to the Platform Analytics node.

The FLEXnet Manager Reporting data source is named `FNMReportDB`.

  - In UNIX, run `ANALYTICS_TOP/1.2/bin/dbconfig.sh add FNMReportDB`
  - In Windows, run `ANALYTICS_TOP/1.2/bin/dbconfig add FNMReportDB`
4. Modify the FLEXnet Manager data loader configuration file to point to the FLEXnet Manager servers.
  - a) Edit the FLEXnet Manager data loader configuration file.
    - UNIX: `ANALYTICS_TOP/conf/data loader/fnmloader.properties`
    - Windows: `ANALYTICS_TOP\conf\data loader\fnmloader.properties`
  - b) Change the `DataSource` parameter to specify the name of the data source for the FLEXnet Manager Reporting database.

Each FLEXnet Manager server has its own data source, and each data loader can only access one data source; therefore, for each FLEXnet Manager server you need to specify a separate data source for each data loader.

For the FLEXnet Manager Reporting database,

```
DataSource=FNMRptDB
```

5. Enable the FLEXnet Manager data loader in your cluster.
  - a) Edit the loader controller configuration file for the FLEXnet Manager data loader:
    - UNIX: `ANALYTICS_TOP/conf/plc/plc_license.xml`
    - Windows: `ANALYTICS_TOP\conf\plc\plc_license.xml`
  - b) Enable data gathering for the FLEXnet Manager data loader (fml oader) by modifying the `Enable` attribute of the `<DataLoader Name="fml oader" ... />` element.

The element should now resemble the following:

```
<DataLoader Name="fml oader" ... Enable="true" ... />
```

6. Start the Platform Analytics node host by restarting the loader controller (plc) service.

```
perfadmin stop plc
```

```
perfadmin start plc
```

## Install and configure the Platform FLEXnet Manager scripts

You can only configure Platform's FLEXnet Manager scripts if you configured Platform Analytics to collect FLEXnet license data.

Check that the following are installed and functional:

- FLEXnet Manager, version 11 or later
- Jasper reports

Configure Platform's FLEXnet Manager (FNM) scripts to work with the FLEXnet Manager data loaders and your LSF cluster.

1. Enable the Platform FLEXnet Manager data loader.
  - a) Edit the loader controller configuration file for license data loaders.

Edit `PERF_TOP/conf/plc/plc_license.xml`.

- b) Enable the FLEXnet Manager data loader.

```
<DataLoader Name="fml oader" ... Enable="true" ... />
```

- c) Restart the plc service for your changes to take effect.

2. Deploy the esub script.

This script provides the cluster name and job submission time to the `LM_PROJECT` environment variable, which passes the data to FLEXnet Manager to be collected by the Platform Analytics data loader.

- a) If you have an existing script in place, add the following information to it:

```
modenv(LM_PROJECT => substr($PROJECT_NAME, 0, 5) . ", $LSF_CLUSTER_NAME, $SUBMIT");
```

Note the following:

- `LSF_CLUSTER_NAME` is the name of the LSF cluster where the job is running

## Configure the Platform Analytics node

- `SSUBMIT` is the job submission time as the number of non-leap seconds from 00:00:00 UTC, January 1, 1970.
  - b) If you do not have an existing script, you need to copy and deploy the `esub.fnm` script, which is located in the `PERF` samples directory:
    - UNIX: `$PERF_TOP/samples`
    - Windows: `%PERF_TOP%\samples`
  - 3. Deploy the job starter script.

This script provides the job ID and job array index to the `LM_PROJECT` environment variable, which passes the data to FLEXnet Manager to be collected by the Platform Analytics data loader.

  - a) Edit `lsub_queues` and add the `JOB_STARTER` parameter with the path to the `jstart` script to each queue that you want to control a submitted.

```
JOB_STARTER = /path_to_jstart/jstart
```
  - b) In the command console, reconfigure the master host to activate this change.  
**badmin reconfig**
  - c) In the command console, display detailed queue information to verify the configuration.  
**bqueues -l queue\_name**

For example, if you added the script to the Normal queue,  
**bqueues -l normal**
  - d) If you have an existing script in place, add the following information to it:

```
LM_PROJECT=SLM_PROJECT, SLSB_JOBID, SLSF_JOBINDEX  
export LM_PROJECT
```
  - e) If you do not have an existing script, you need to copy and deploy the `jstarter.fnm` script, which is located in the `PERF` samples directory:
    - UNIX: `$PERF_TOP/samples`
    - Windows: `%PERF_TOP%\samples`
4. Deploy the FNM data loader to a data collection host.

## Enable and configure the remoting server client (asynchronous data loading mode only)

If you intend to use the asynchronous data loading mode, enable and configure the remoting server client.

You should only use the asynchronous data loading mode for sending data from the Platform Analytics node to the database over a slow or unstable network.

1. Log into the Platform Analytics node host.
2. Navigate to the `ANALYTICS_TOP/conf` directory.
3. Rename the `remotingclient.TMP` file to `remotingclient.xml`.
4. Edit the `remotingclient.xml` file and change the `serverBindAddress` and `serverBindPort` parameters according to the remoting server configuration.
5. Edit the `perf.conf` file and add the following parameters:

```
ASYNC_LOADING_ENABLED=Y  
ASYNC_LOADING_MODE=JBossRemoting
```
6. Restart the loader controller service (`plc`).

```
perfadmin stop plc
```

```
perfadmin start plc
```

Configure the Platform Analytics node

## Optimize the Platform Analytics server

Modify the Platform Analytics server to enhance performance.

# What you need to do

The following is optional.

1. [Change the data retention period](#) on page 48
2. [Split the data transformer tasks to disperse workload](#) on page 49

## Change the data retention period

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

The data purger consists of multiple scheduled tasks (Parti ti onMai ntenanceGroup\*), which are enabled by default.

1. Launch the vsql command line.
  - a) Navigate to the bin subdirectory of the Vertica installation directory.

By default, this is /opt/vertica/bin.

- b) Run vsql to connect to the database.

```
./vsql -d database_name -p port -U username -w password
```

where

- *database\_name* is the name of the database
- *port* is the TCP port number or the local socket file extension in which the server is listening for connections. The default is port number 5433.
- *username* is the name of the user with which to connect to the database, instead of the default user (the database administrator).
- *password* is the password for the database user.

Alternately, you can run vsql with no options to accept the defaults and specify the administrator password at the prompt.

2. Examine the current data retention periods of the database tables for Platform Analytics.
  - To examine the retention periods for all Platform Analytics database tables, run the following from the vsql command line:

```
SELECT TABLE_NAME, DATA_DAYS_RANGE  
FROM SYS_TABLES_TO_PARTITION;
```

- To examine the retention periods for a specific database table, run the following from the vsql command line:

```
SELECT TABLE_NAME, DATA_DAYS_RANGE  
FROM SYS_TABLES_TO_PARTITION  
WHERE TABLE_NAME='table_name';
```

where *table\_name* is the name of the table you want to examine.

The output displays the name of the table and the corresponding data retention period in days.



For example, to view the data retention period for the RESOURCE\_METRICS\_BUILTIN table, run the following from the vsql command line:

```
SELECT TABLE_NAME, DATA_DAYS_RANGE
FROM SYS_TABLES_TO_PARTITION
WHERE TABLE_NAME='RESOURCE_METRICS_BUILTIN';
```

3. Change the data retention period for the appropriate database tables.
  - a) For each database table to change, run the following from the vsql command line:

```
UPDATE SYS_TABLES_TO_PARTITION
SET DATA_DAYS_RANGE='retention_period'
WHERE TABLE_NAME='table_name';
```

where

- *retention\_period* is the new retention period, in days
- *table\_name* is the name of the table you are changing

- b) Commit the changes to the database.

Run the following from the vsql command line:

```
COMMIT;
```

For example, to change the data retention period of the RESOURCE\_METRICS\_BUILTIN table to 2192 days, run the following from the vsql command line:

```
UPDATE SYS_TABLES_TO_PARTITION
SET DATA_DAYS_RANGE='2192'
WHERE TABLE_NAME='RESOURCE_METRICS_BUILTIN';
COMMIT;
```

## Split the data transformer tasks to disperse workload

By default, there are two 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 five tasks. The examples will make use of this table (specifically, with Task 1).

Task	Data transformer name	Data flow entry
1	ClusterCapacity	main_cluster_capacity.xml
2	WorkloadStatistics	main_workload_statistics.xml
3	FlexlmLicenseUsage	main_flexlm_licusage.xml
4	Hardware	main_hardware.xml
5	Jobmart	main_jobmart.xml

1. Log into the Platform Analytics server host.

2. Create and enable a new scheduled task in the Platform Analytics Console.
  - a) Launch the Platform 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 1 from the table with the recommended format of splitting data transformers, specify the following:

    - Scheduled Task: Specify Task1 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 1 on a UNIX host,

```
cd ANALYTICS_TOP/tasks
```

```
mkdir Task1
```

```
cd Task1
```

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

For example, create `task1.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 1, the `task1.tsk` file should contain the following lines:

```
# Cluster Capacity  
datatransformer/flow/clustercapacity/main_cluster_capacity.xml
```

## Optimize the Platform Analytics node

Optional. Modify the Platform Analytics node to enhance performance.

## What you need to do

1. [Increase JVM memory](#) on page 52
2. [Disperse the Platform Analytics node workload](#) on page 52
3. [Optimize specific data loaders](#) on page 55

## Increase JVM memory

Increase the Java Virtual Machine (JVM) memory to at least 2 GB for the Platform 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 Platform 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 Platform Analytics node workload

You can reduce performance issues by dispersing the Platform 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"><li>• <code>plc_ego.xml</code></li><li>• <code>plc_coreutil.xml</code></li></ul>
Job-related	<ul style="list-style-type: none"><li>• <code>plc_lsf.xml</code></li><li>• <code>plc_bj_obs-sp012.xml</code></li></ul>
Advanced job-related	<ul style="list-style-type: none"><li>• <code>plc_lsf_advanced.xml</code></li></ul>
License-related	<ul style="list-style-type: none"><li>• <code>plc_license.xml</code></li></ul>

- [Disperse the Platform Analytics node workload to multiple hosts](#) on page 53

Disperse the workload to multiple hosts if your Platform Analytics node is experiencing performance issues and cannot handle all the workload by itself.

- [Disperse the Platform Analytics node workload to multiple loader controllers in one node](#) on page 53

Disperse the workload to multiple loader controllers within a host to reduce performance bottlenecks if your Platform 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 Platform 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 Platform 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 Platform 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 Platform Analytics node, disable the data loading categories that the node will not be handling.
  - a) Navigate to the `ANALYTICS_TOP/conf/plc` directory.
  - b) Move or rename the configuration files for the data loader categories that the Platform Analytics node will no longer handle.

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

## Disperse the Platform 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 `plc_job` to handle job-related data loaders, `plc_advanced_job` to handle advanced job-related data loaders, and `plc_license` to handle license-related data loaders, while the original loader controller (`plc`) 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_ls.xml` and `plc_bj_obs-sp012.xml` from `ANALYTICS_TOP/conf/plc` to `ANALYTICS_TOP/conf/plc_job`.
2. Move `plc_ls_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 (`f1exlicusageloader`) 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 Platform Analytics node host that is running the FLEXnet usage data loader.
- b) Edit the `ANALYTICS_TOP/conf/dataloader/f1exlicusage.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 (`lsfeventsloder`) 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 Platform 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
...
End Parameters
```

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

```
lsadmin reconfig
```

Optimize the Platform Analytics node



## Remove Platform Analytics software from the host

Remove Platform Analytics from a UNIX or Windows host.

## What you need to do

- [Remove Platform Analytics software from a UNIX host](#) on page 58
- [Remove Platform Analytics software from a Windows host](#) on page 58

## Remove Platform Analytics software from a UNIX host

1. Log into the Platform Analytics host with the same account that you used to install Platform Analytics.
2. Navigate to the *ANALYTICS\_TOP/uninstall* directory.
3. Run the uninstall package.

**`./uninstaller.bin`**

## Remove Platform Analytics software from a Windows host

1. Log into the Platform Analytics host with the same account that you used to install Platform Analytics.
2. Run the Platform Analytics uninstaller.
  - a) Select Start > Settings > Control Panel > Add/Remove Programs.
  - b) Select your Platform Analytics installation and click Remove.
  - c) If you receive a message that the program has already been uninstalled, manually run the Platform Analytics uninstaller.
    1. Navigate to the *ANALYTICS\_TOP\uninstall* directory.
    2. Run the uninstall package.

**`uninstaller.exe`**

Restart your host to complete the software removal process.

---

# Index

## A

- Analytics Console 38
- ANALYTICS\_TOP 25, 29
- architecture 8

## D

- data loaders
  - log files 29
  - verifying 29
- data sources
  - FNMReportDB 42
- database
  - Vertica
    - schema 20
- display environment
  - setting 24, 28, 42

## E

- environment
  - display
    - setting 24, 28, 42
  - LSF
    - sourcing 28, 40, 41
  - PERF
    - sourcing 29

## events

- Analytics Console 38
- notification 25, 38

## F

### files

- installation
  - Analytics 18
  - database 18
  - ROLAP 18

- license 11, 18

## FLEXnet

- data loaders
  - configuring 41, 42
  - enabling 41, 42
- data source 42

- FNMReportDB 42

## J

- JDBC driver 28

## L

- license 11
  - FLEXnet
    - data loaders 41, 42
- loader controller
  - ports 29
- LSF\_ENVDIR 29

## N

- node 5
  - Analytics Console 38
  - ports 6
  - uninstalling 58

## P

- PERF
  - services
    - stopping 40
- ports 6
  - loader controller 29
  - node 6
  - server 6, 25

## S

- scheduled tasks 38
- server 5
  - configuring 38
  - ports 6, 25
  - uninstalling 58
  - verifying 38

- services
  - stopping 40
- source environment
  - LSF 28, 40, 41
  - PERF 29
- system architecture 8

## U

- uninstall 58

## V

- Vertica
  - database schema 20
  - JDBC driver 28