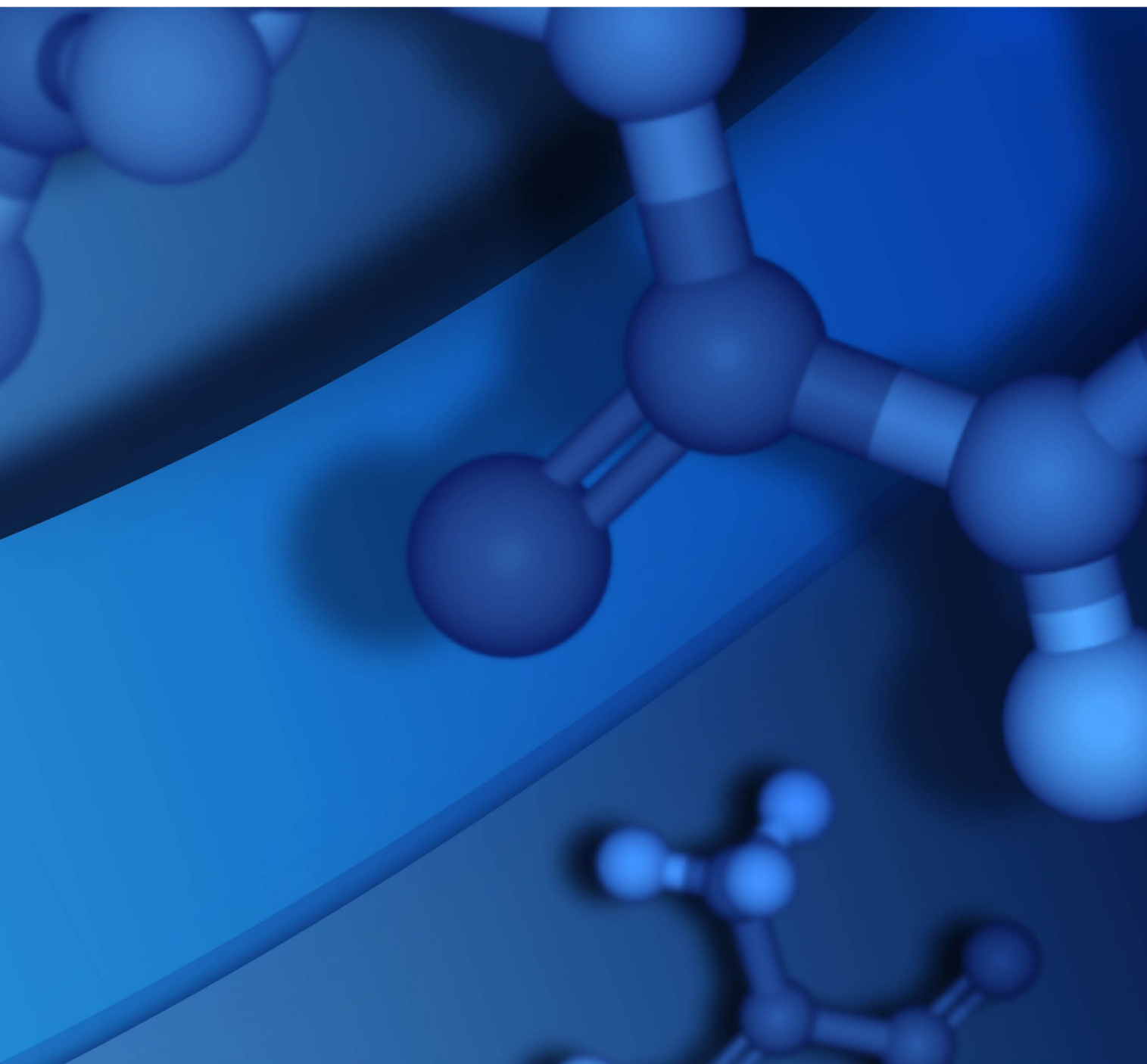


INSTALLATION GUIDE

PIPELINE PILOT SERVER 2021



Copyright Notice

©2020 Dassault Systèmes. All rights reserved. 3DEXPERIENCE, the Compass icon and the 3DS logo, CATIA, SOLIDWORKS, ENOVIA, DELMIA, SIMULIA, GEOVIA, EXALEAD, 3DVIA, 3DSWYM, BIOVIA, NETVIBES, IFWE and 3DEXCITE, are commercial trademarks or registered trademarks of Dassault Systèmes, a French "société européenne" (Versailles Commercial Register # B 322 306 440), or its subsidiaries in the U.S. and/or other countries. All other trademarks are owned by their respective owners. Use of any Dassault Systèmes or its subsidiaries trademarks is subject to their express written approval.

Acknowledgments and References

To print photographs or files of computational results (figures and/or data) obtained by using Dassault Systèmes software, acknowledge the source in an appropriate format. For example:

"Computational results were obtained by using Dassault Systèmes BIOVIA software programs. Pipeline Pilot Server was used to perform the calculations and to generate the graphical results."

Dassault Systèmes may grant permission to republish or reprint its copyrighted materials. Requests should be submitted to Dassault Systèmes Customer Support, either by visiting <https://www.3ds.com/support/> and clicking **Call us** or **Submit a request**, or by writing to:

Dassault Systèmes Customer Support
10, Rue Marcel Dassault
78140 Vélizy-Villacoublay
FRANCE

Contents

Chapter 1: Pipeline Pilot Server 2021 Installation	1
BIOVIA License Files	1
Using Ansible Software	1
Additional Information	1
Chapter 2: Getting Started	2
Overview	2
Dassault Systèmes Support Resources	2
Security Considerations	2
Deployment Options	2
Managing Multiple Installations	3
Side-by-side installations	3
Protocol Database Compatibility	3
Port Management	3
Sharing an XMLDB with Multiple Servers	4
Moving or Migrating an Installation	4
Preparing to Migrate	4
Migrating a Pipeline Pilot Configuration	4
Migration Information for Pipeline Pilot Directories	6
Chapter 3: Pipeline Pilot Services	8
Overview	8
Shutting Down Services on Windows	8
Shutting Down Apache on Linux	9
Chapter 4: Upgrading Pipeline Pilot	10
Upgrade Order	10
Upgrading Pipeline Pilot on Windows	10
Pre-upgrade steps	10
Upgrade steps	10
Upgrading Pipeline Pilot on Linux	11
Pre-upgrade steps	11
Upgrade steps	11
Chapter 5: Installing Pipeline Pilot	13
Pre-Installation Checklist for Windows	14
Installing on Windows	17

Windows Guidelines	18
Windows Requirements for Load Balancing	18
Local Job Temporary Directory	18
Impersonation on Windows	18
Pre-Installation Checklist for Linux	20
Installing on Linux	23
Linux Guidelines	24
Installation paths	24
Red Hat Linux	24
Local Authentication and SELinux	24
Distributed File System and NFS Configuration	24
Clock Synchronization	26
Fixing Java JDK Timezone Issues	26
Possible Interaction with /usr/sbin/tmpwatch	26
Chapter 6: Post-Installation	27
Post-Installation Steps	27
Post-Install Checklist for Linux	30
Testing Client Software on Linux	32
Server Configuration Checklist	32
Pipeline Pilot Server Home Page	34
Pipeline Pilot Administration Portal	35
Installing Pipeline Pilot Client	36
Setting Foundation Hub as the Authentication Server	36
Migrating Custom Pipeline Pilot Authentication Data to Foundation Hub	37
Chapter 7: Uninstalling Pipeline Pilot	39
Before Uninstalling	39
Uninstalling on Windows	39
Uninstalling from a Linux Server	40
Chapter 8: Load Balancing Pipeline Pilot	42
Load Balancing Server Requirements	42
Application Specific Customizations	42
Preparing Servers for Load Balancing	43
Creating Required Folders	43
General Approach	44
Selecting Pipeline Pilot Administration Settings	44

Setting the Pipeline Pilot Service User (Windows)	44
Configuring a Primary Pipeline Pilot Server for Load Balancing	45
Setting up the Registry Keys (Windows)	45
Configuring Certificate Information	46
Pipeline Pilot Server Administration Configuration	47
Configuring Subsequent Pipeline Pilot Servers for Load Balancing	48
Setting up the Registry Keys (Windows)	48
Pipeline Pilot Server Administration Configuration	49
Testing the Load Balanced Configuration	49
Chapter 9: Installing on Grids and Clusters	51
Pipeline Pilot Clustering Overview	51
Clustering Modes	51
Distributed Grid Computing Overview	51
Parallel Computing	52
Pipeline Pilot Servers with Grid-enabled Clusters	52
Requirements	52
Requirements for Clustering	52
Best Practices	53
Requirements for Grid Engines	53
Pre-Installation for a Grid or Cluster	53
User Accounts	53
NFS Configuration	54
Reducing NFS Dependencies for Grids and Clusters	54
Security	55
Grid Engine Setup	55
Pre-Installation Checklist	55
Installation Guidelines	57
Post-Installation for a Grid or Cluster	57
Configuring your Linux Server	58
Configuring a Cluster	58
Testing the Cluster	59
Configuring a Grid	59
Configuring Nodes	59
Defining Grid Cluster Queues for Nodes	60
Configuring the Server	60

Custom Grid Engines	61
Impersonation	61
Testing a Grid Engine	61
Appendix A: Unattended Installation	62
Setting Required Pre-Installation Environment Variables	62
Installation Command Syntax	63
Package Names	64
Installation Command Example	66
Appendix B: Required RPMs	67
Red Hat 8.1+ and CentOS 8.1+	67
Red Hat 7.5+ and CentOS 7.5+	67
Appendix C: Installing GPL-licensed Biology Tools	69
Windows	69
Linux	69
Appendix D: Installing BIOVIA License Pack for Materials Studio Collection	71
Linux Post-Installation Instructions	71
Dassault Systèmes Support Resources	72

Chapter 1:

Pipeline Pilot Server 2021 Installation

BIOVIA License Files

To install Pipeline Pilot Server and collections, you need a valid license file. Instructions are included with the license file for activating a new license.

Licenses are made available when the software is purchased. If you need further assistance relating to licenses you should initially contact your account manager.

Using Ansible Software

BIOVIA supports using Ansible to manage installations and upgrades for Pipeline Pilot, Foundation Hub, CISPro, Compose and Capture, and Workbook. For details, see the *Ansible Installation Guide for BIOVIA*. This document is included in the zipped product documentation for ULM, which is available from the Dassault Systèmes Download Platform alongside the product installers.

Additional Information

- For information about supported hardware and software, see the *Pipeline Pilot System Requirements* document.
- For details about advanced server deployments, see the *Pipeline Pilot Deployment Guide*.
- For details about installing R Software, see the *R Software for Pipeline Pilot Server 2021 Installation and Configuration Guide*.
- For information about installing BIOVIA Foundation Hub see the *Foundation Hub Installation Guide*.
- For more information about Pipeline Pilot 2021 and other Pipeline Pilot software products, visit [Dassault Systèmes User Communities](#).

Chapter 2:

Getting Started

Overview

Upfront planning is critical to a successful enterprise deployment. See the *Pipeline Pilot Deployment Guide*, then use the pre-installation checklists and related information in this chapter to plan your deployment. The checklists apply to most deployments.

IMPORTANT!

- Pipeline Pilot Server 2021 requires Pipeline Pilot Client 2021. Users must upgrade their client in order to access the server.
- Some applications and collections must be upgraded separately (for example, Insight and Insight for Excel). Before installing Pipeline Pilot, ensure that you have access to the separate installers that are compatible with this version of Pipeline Pilot Server.
- BIOVIA Draw does not upgrade automatically when you upgrade from an earlier version of Pipeline Pilot Server. You will need to upgrade Draw separately.

Dassault Systèmes Support Resources

For additional resources or to contact Dassault Systèmes Customer Support, visit the Support portal:

<https://www.3ds.com/support/>

From this portal, you can:

- Call or email Dassault Systèmes Customer Support
- Submit a request
- Download installers
- Access hardware and software requirements
- Access Knowledge Base
- Access Communities and Twitter feeds

Security Considerations

Pipeline Pilot supports various configuration options and settings that can help protect your data from security vulnerabilities. When planning your deployment, review the recommendations for internet-accessible deployments and other environments that require a high level of security.

See Security Considerations in the *Pipeline Pilot Deployment Guide*.

Deployment Options

There are several advanced options for deploying Pipeline Pilot including:

- Workstation
- Standalone
- Load balancing: For more information, see [Load Balancing Pipeline Pilot](#) on page 42.
- Clustering: For more information, see [Installing on Grids and Clusters](#).
- Distributed grid computing: For more information, see [Installing on Grids and Clusters](#).

Managing Multiple Installations

Pipeline Pilot supports side-by-side installations of new and existing products. You can run Pipeline Pilot 2021 and previous versions on the same machine. With side-by-side installations, the installer can copy components and protocols from an existing protocol database (XMLDB) into the new installation. You can also upgrade an existing installation to the latest version.

Side-by-side installations

To run different versions side-by-side, each version needs to be installed in a unique path or drive on your server. They cannot all reside in the same directory or subdirectories. You also need to run side-by-side versions of the clients that correspond to each server installation, assuming they are mutually incompatible versions.

On a Windows server, the default paths are:

Version	Default Install Path
Pipeline Pilot Server 2021, 2017 R2, 2017, 2016, 9.5	C:\Program Files\BIOVIA\PPS
Pipeline Pilot Server 9.0, 9.1, and 9.2	C:\Program Files\Accelrys\AEP
Pipeline Pilot Server 8.x	C:\Program Files\Accelrys\PPS
Pipeline Pilot Server 7.5	C:\Program Files\Accelrys\SES
Pipeline Pilot 7.0 and earlier versions	C:\Program Files\SciTegic (or SciTegic 7.0 if this location already contains an installation)

Protocol Database Compatibility

The Pipeline Pilot Client supports connecting to multiple XMLDBs through the **Network** tab. Pipeline Pilot Clients after version 8.5 can connect to databases of previous software versions, but can only save files to the current database of a Pipeline Pilot Server with the same version.

Port Management

The following port numbers used by Pipeline Pilot:

Type	Default Port
HTTP	9944
HTTPS	9943
Tomcat Shutdown	9945
Tomcat HTTP	9946
Jupyter	9947

You can have side-by-side installations of Pipeline Pilot 2021 on the same server where earlier versions are installed. Each installation requires unique ports. If the default ports listed above are already in use by an earlier version of Pipeline Pilot Server, use alternate port numbers for Pipeline Pilot Server 2021.

Sharing an XMLDB with Multiple Servers

Pipeline Pilot Clients running at multiple sites can access the same protocols and components if they share the same protocol database (XMLDB). To provide this functionality, the computational servers that process protocol jobs at each site need to communicate with one server that operates as the XMLDB for all the other servers. Clients are only aware of the computational server where they run protocols. They have no information about the machine hosting the XMLDB. See [Moving or Migrating an Installation](#) for more information about XMLDB.

Moving or Migrating an Installation

You can move or migrate your server location by transferring the Pipeline Pilot configuration and certain directories to the new server.

After uninstalling, some files remain intact, making it possible for you to relocate existing data to a server re-installation. Specifically, your XMLDB remains intact in the `<pps_install>` folder, so you can easily move or migrate it to a new location. However, you should *not* uninstall the server being migrated until you have completed these steps.

Preparing to Migrate

You should export the Pipeline Pilot Admin Portal configuration and all data sources from the server being migrated.

1. Open the **Maintenance > Export Import Configuration** page of the Pipeline Pilot Admin Portal on the server being migrated and click **Export Configuration**.

Ensure that the generated configuration file is available from the server you are migrating to.

2. Open the **Setup > Data Sources** page of the Pipeline Pilot Admin Portal and click **Export All Data Sources**.

Ensure that the generated data source file is available from the server you are migrating to.

Migrating a Pipeline Pilot Configuration

To migrate a Pipeline Pilot Server configuration you need to move some files and folders manually and import the Pipeline Pilot Admin Portal configuration and data sources.

IMPORTANT! `<pps_install>` is the root of the Pipeline Pilot installation.

On Windows this is typically:

C:\Program Files\BIOVIA\PPS

On Linux this depends on the location of your Pipeline Pilot installation, but may be:

[Home]/BIOVIA/PPS

1. Open the **Setup > Folder Locations** page of the Pipeline Pilot Admin Portal for your new server and check that the **User Directory** is set to the default value, on Windows this is typically C:\PPS\public\users.
2. Manually migrate the following folders from the location of your previous, now uninstalled, Pipeline Pilot Server to your new Pipeline Pilot Server installation, even if it is on a different machine:
 - `<pps_install>/xml/db/Users`
 - `<pps_install>/xml/db/Components`
 - `<pps_install>/xml/db/Protocols`

Do not attempt to migrate the `<pps_install>/xml/db/Objects` folder contents.

- Optionally migrate the following folders and resources to your new Pipeline Pilot Server:

- `<pps_install>/apps/<vendor>`

Where `<vendor>` is *not* **scitegic**.

You will need to reinstall each package on your new Pipeline Pilot Server using the `pkgutil -r` command. See the *Pipeline Pilot Application Packaging Guide* for more information on using `pkgutil`.

- `<pps_install>/web/apps`
- `<pps_install>/apps/scitegic/web`
`<pps_install>/web/webport/<custom_name>`

These directories may contain customized Web Port features.

You will need to deploy these customizations on your new Pipeline Pilot Server using the `customize_webport` script. See the *Pipeline Pilot Web Port Customization Guide* for more information on using `customize_webport`.

- `<pps_install>/public/caches`
- `<pps_install>/public/drivers/jdbc`
- `<pps_install>/licensing`

- Import your data sources from the previous Pipeline Pilot Server into the Pipeline Pilot Admin Portal for your new server.

Open the **Setup > Data Sources** page of the Pipeline Pilot Admin Portal and click **Import Data Source**. Import the data source file exported from the server being migrated.

Note: If you have any DSN databases you should either add these as Pipeline Pilot data sources or recreate the definitions on the new server in the ODBC Data Source Administrator tool in Windows. On Linux migrate the file :
`<pps_install>/apps/scitegic/core/packages_linux64/datadirect/odbc.ini`

- Open the **Maintenance > Export Import Configuration** page of the Pipeline Pilot Admin Portal on the server being migrated and click **Import Configuration**. Select the configuration file you exported from your previous server.
- Activate the migrated content on your new server:

- **Windows:** Run the Pipeline Pilot installer and select the **Reinstall/Repair** option.
- **Linux:** Run the Pipeline Pilot installer and select the **Reinstall an existing installation** option. Then run `scirootinstall` as root to configure autostart, PAM configuration, and optionally turn on impersonation. See [Post-Install Checklist for Linux](#) for more information.

If your protocols are not available in the Pipeline Pilot Client for the new server after reinstallation you may need to for the protocol database to be reindexed:

- **Windows:**
`cd <pps_install>\bin`
`Dbutil.exe -s`
- **Linux:**
`cd <pps_install>/linux_bin`
`source ./ppvars.sh`
`./Dbutil -s`

- Verify that your migrated installation and configuration are functioning as expected.
- Uninstall Pipeline Pilot from your previous server.

Migration Information for Pipeline Pilot Directories

Default Directory	Description	When moving you server to a new location, do this:
<pps_install>/xmldb	Protocol database that stores protocols and components specific to the server, including version history and custom protocols. Some files are public (Components and Protocols subfolders); others are owned by specific users (User subfolder).	Copy this directory to the target location, so users can access their protocols, components, and version history.
<pps_install>/xmldb/objects	Server startup configuration. This information should only be located where Pipeline Pilot is installed. Contents are configured on your server during installation and must always reside in the default directory.	Do not copy files from the objects folder to the target server. Instead, use the Admin Portal to customize your server settings on the target server. This ensures that all registry files are correctly configured.
<pps_install>/web	Custom web applications, Web server customizations, job folders.	When moving your server to a new location, you need these files on the target server, so users can continue to run your web applications.
<pps_install>/apps/scitegic	This directory is removed, but any other non-BIOVIA packages in /apps are not deleted.	No action
<pps_install>/licensing	License-related files.	When moving your server to a new location, copy this folder to the target location. During installation, specify the path to your license file to install the appropriate packages.
<pps_install>/public	If this directory contains custom user data, including /public/users, it is not removed.	No action
<pps_install>/logs	Log files.	You do not need to copy these files to other servers as new log files are created after installing. Move this folder to preserve the server history, (for usage tracking purposes).
<pps_install>/install	Contains installation files.	You do not need to copy these files

Default Directory	Description	When moving you server to a new location, do this:
		to other servers.

Chapter 3:

Pipeline Pilot Services

Overview

All Pipeline Pilot services (including Apache HTTPD and Java or Tomcat) need to be shut down before you can install a new version of Pipeline Pilot on your server. This frees up the ports, which is necessary for a successful installation.

The default port numbers used by the Apache web service are:

Type	Default Port
HTTP	9944
HTTPS	9943
Tomcat Shutdown	9945
Tomcat HTTP	9946
Jupyter	9947

The following services are installed with your server installation on Windows:

- BIOVIAPipeline Pilot <version> (Manager): Manages the lifetime of the Pipeline Pilot Server environment.
- BIOVIAPipeline Pilot <version> Service (Httpd): Runs the Apache web server.
- BIOVIAPipeline Pilot <version> Service (Jupyter Notebook): Runs the Jupyter server.
- BIOVIAPipeline Pilot <version> Service (Tomcat): Runs the Tomcat Java application server.

Notes:

- The installer can shut down these services for you while you perform a new Pipeline Pilot installation. You can also use the Windows Services console to shut down services the same way you shut down web services with earlier Pipeline Pilot Server releases.
- For Pipeline Pilot Server 8.5 and earlier, you need to manually shut down the Apache web server before you can perform a new installation. If multiple Pipeline Pilot servers are running simultaneously, multiple Apache services are also running on your server and it is necessary to shut down *all* currently running services.

Shutting Down Services on Windows

1. From Windows Control Panel, go to **Administrative Tools > Services**. The Services console opens.
2. To shut down all services, right click the **BIOVIA Pipeline Pilot [version] (Manager)** service in (or other service for an earlier version) and select **Stop**.

As the manager service, stopping this service will also stop the Httpd and Tomcat services.

To shut down either the Httpd or Tomcat services individually, right-click on the appropriate entry and select **Stop**.

3. If you want to ensure that all ports are released, open a command window and run the following:
`netstat -a`

A list of port numbers currently in use is displayed. This list should not include Apache ports.

Tip: It can take a few minutes for Apache to release the ports. Retype the command to refresh the list.

Shutting Down Apache on Linux

1. Change directories to your <pps_install> directory:

```
$ cd linux_bin
```

2. Stop the server:

```
$ ./stopserver
```

3. Ensure that the ports are released:

```
$ netstat -lt
```

A list of port numbers currently in use is displayed. The default Apache port numbers used by Pipeline Pilot should not be listed.

Tips:

- It might take a few minutes for Apache to release the ports. To refresh the list, retype the command.
- You can use the following command to check for running HTTPD processes in the scitegic directory:

```
$ ps -aux | grep scihttpd
```

Chapter 4:

Upgrading Pipeline Pilot

Upgrade Order

1. Upgrade Foundation Hub.
2. Upgrade Pipeline Pilot.
3. Upgrade remaining BIOVIA applications.

Upgrading Pipeline Pilot on Windows

Pre-upgrade steps

1. Notify users that the services for BIOVIA applications will be stopped and restarted during the upgrade. Ask them to log out of all BIOVIA applications including Pipeline Pilot Client and refrain from using them until after the upgrade.
2. For load balanced environments, perform the steps for the upgrade on each Pipeline Pilot Server in the load balanced configuration.
3. Before upgrading, check that you can log into your current version of Pipeline Pilot.
4. From the Admin Portal, ensure that there are no jobs in Running or Queued state.
5. Clean up your jobs folder. For more information, see the Pipeline Pilot Help Center: **Admin > Status and Monitoring > Job Management > Job Folder Maintenance**.
6. Stop the services:
 - Stop the services for all BIOVIA applications.
 - Stop the BIOVIA Foundation Hub service.
 - Stop the Pipeline Pilot services.
 - If you are using BIOVIA applications that rely on Pipeline Pilot, you must stop the services for these applications **before** you upgrade Pipeline Pilot.

Upgrade steps

1. Log on to the destination computer as an administrator.
2. Extract the BPP2021.7z file.
3. In the extracted folder, double-click `\bin\scitegicsetup.exe` to run the installer.
4. When prompted, choose from one of the following installation options:
 - **Install to an existing location. Upgrade any existing protocols:** Replace an existing installation with this new version and all your protocols will be upgraded. The port numbers and installation directory from the previous version will be maintained.
 - **Install to a new location. Do not upgrade any protocols from any existing installations:** Create a new installation directory and does not copy or upgrade from a previous version.
 - **Install to a new location. Copy protocols from selected installation to new location and upgrade them:** If you have custom protocols, it is recommended that you choose this option so that you can validate your protocols on the new platform before removing the previous versions. Custom protocols in the existing installation are copied and upgraded to work in the

Pipeline Pilot Server server. This option will also copy administration settings from the previous installation.

5. In the **Select product(s) to install** screen, choose to **Backup the XMLDB**.
6. Follow the remaining prompts until the upgrade is complete.

Upgrading Pipeline Pilot on Linux

Pre-upgrade steps

1. Notify users that the services for BIOVIA applications will be stopped and restarted during the upgrade. Ask them to log out of all BIOVIA applications including Pipeline Pilot Client and refrain from using them until after the upgrade.
2. For load balanced environments, perform the steps for the upgrade on each Pipeline Pilot Server in the load balanced configuration.
3. Before upgrading, check that you can log into your current version of Pipeline Pilot.
4. From the Admin Portal, ensure that there are no jobs in Running or Queued state.
5. Clean up your jobs folder. For more information, see the Pipeline Pilot Help Center: **Admin > Status and Monitoring > Job Management > Job Folder Maintenance**.
6. Stop the services:
 - Stop the services for all BIOVIA applications.
 - Stop the BIOVIA Foundation Hub service.
 - Stop the Pipeline Pilot services.
 - If you are using BIOVIA applications that rely on Pipeline Pilot, you must stop the services for these applications **before** you upgrade Pipeline Pilot.
7. Backup the XMLDB:


```
. <pps_install>/linux_bin/ppvars.sh
<pps_install>/linux_bin/dbutil -backup -file xmldb_backup.zip.
```

Upgrade steps

1. Extract the contents from the BPP2021.tgz file:


```
tar -xvzf
```
2. If you are using Red Hat Enterprise Linux and have Security-Enhanced Linux enabled, see [Local Authentication and SELinux](#).
3. Run `sciinstall` as the non-root user you created in [Pre-Installation Checklist for Linux](#):


```
$ ./sciinstall
```
4. When prompted, choose from one of the following installation options:
 - **[1] Install a new server:** Create a new installation directory and does not copy or upgrade from a previous version. If a previous installation exists, specify different port numbers and installation directory.
 - **[2] Reinstall an existing installation:** Repair an existing installation.
 - **[3] Upgrade an existing installation:** Replace an existing installation with this new version and all your protocols will be upgraded. The port numbers and installation directory from the previous version will be maintained.

- **[4] Install a new server to a new location and upgrade the XMLDB database from existing installation:** If you have custom protocols, it is recommended that you choose this option so that you can validate your protocols on the new platform before removing the previous versions. Custom protocols in the existing installation are copied and upgraded to work in the Pipeline Pilot Server server. This option will also copy administration settings from the previous installation. Note that this option will only upgrade the XMLDB database and NOT any other databases.
5. Follow the remaining prompts until the upgrade is complete.

Chapter 5:

Installing Pipeline Pilot

This section describes how to install Pipeline Pilot according to your operating system and details the pre-installation steps that are required.

Pre-Installation Checklist for Windows	14
Installing on Windows	17
Pre-Installation Checklist for Linux	20
Installing on Linux	23

Pre-Installation Checklist for Windows

To ensure that your Pipeline Pilot installation goes smoothly, complete the checklist below.

IMPORTANT! If your deployment includes Foundation Hub, you must install BIOVIA Foundation Hub before installing Pipeline Pilot Server.

- If you are installing on Linux, see [Pre-Installation Checklist for Linux](#) and [Installing on Linux](#).
- If you are upgrading, see [Chapter 4: Upgrading Pipeline Pilot](#).

IMPORTANT! If you are installing on a machine with an older version of Pipeline Pilot, it is recommended that you backup your components and protocols before installing. See [step 13](#) in the checklist.

Do this	Details	Done
1. Obtain BPP2021.7z.		<input type="checkbox"/>
2. Obtain required system administrator privileges to install software on your server.		<input type="checkbox"/>
3. Locate your BIOVIA license file (*.lic).	<p>A valid license file is current (not expired) and accessible during the installation. For each new license file that you obtain, instructions are available that explain how to update and activate the license file.</p> <p>Licenses are made available when the software is purchased. If you need further assistance relating to licenses you should initially contact your account manager.</p> <div> <p>Note: If you are planning to use the Materials Studio collection, you will need to perform additional installation steps. See Appendix D: Installing BIOVIA License Pack for Materials Studio Collection.</p> </div>	<input type="checkbox"/>
4. Verify that your server meets the hardware and software requirements.	See the <i>Pipeline Pilot System Requirements</i> document.	<input type="checkbox"/>
5. Check to see if any protocols are currently running on your server.	Active jobs on the server must finish running before installing Pipeline Pilot. Cancel jobs in the Admin Portal (Jobs > Running Jobs).	<input type="checkbox"/>
6. Shut down Apache.	Apache cannot be running on the server where you are installing Pipeline Pilot.	<input type="checkbox"/>

Do this	Details	Done												
	See Chapter 3: Pipeline Pilot Services .													
7. Shut down the Java server.	If your server also runs the Java Server package (optional), Apache Tomcat cannot be running when you install Pipeline Pilot. See Chapter 3: Pipeline Pilot Services .	<input type="checkbox"/>												
8. Close all instances of the Admin Portal and Pipeline Pilot Client, and Web Port clients.	Shut down all Pipeline Pilot and all Pro Client applications on the server. Close all Pipeline Pilot clients with current connections to the protocol database (XMLDB) on the server you are about to upgrade.	<input type="checkbox"/>												
9. Disable all virus scanning software and firewalls on the server.	Some virus scanners and firewall applications can slow down or even corrupt the installation.	<input type="checkbox"/>												
10. Verify that you can use the Pipeline Pilot default port numbers with the Apache web server:	<div>If these port numbers are already in use by another web service, you can manually assign different port numbers for Apache during the installation.</div> <table><tr><th>Type</th><th>Default Port</th></tr><tr><td>HTTP</td><td>9944</td></tr><tr><td>HTTPS</td><td>9943</td></tr><tr><td>Tomcat Shutdown</td><td>9945</td></tr><tr><td>Tomcat HTTP</td><td>9946</td></tr><tr><td>Jupyter</td><td>9947</td></tr></table> <div>Decide in advance what port numbers to use. You need to refer to these port numbers when accessing the Pipeline Pilot Home Page and server-related applications. Make a note of any custom port numbers you decide to manually assign. The HTTP and HTTPS port numbers are defined during installation. The other three are automatically selected during installation. However, you can explicitly define all port settings. All use of Tomcat specific ports is within the server node, so they should be externally blocked to increase protection. See the Pipeline Pilot Help Center: Admin > Setup and Configuration > Server Configurations > Reconfiguring Ports.</div>	Type	Default Port	HTTP	9944	HTTPS	9943	Tomcat Shutdown	9945	Tomcat HTTP	9946	Jupyter	9947	<input type="checkbox"/>
Type	Default Port													
HTTP	9944													
HTTPS	9943													
Tomcat Shutdown	9945													
Tomcat HTTP	9946													
Jupyter	9947													

Do this	Details	Done
11. Do you need to run the Apache Web Server under a specific user to extend and control local network resources?	If yes, you need to obtain the specific user account and password.	<input type="checkbox"/>
12. Are you installing on a grid, cluster, web farm, or reverse proxy?	If yes, configure and test your servers to ensure they are fully operational before installing Pipeline Pilot. For details, see the <i>Pipeline Pilot Deployment Guide</i> .	<input type="checkbox"/>
13. Are you installing on a machine with an older version or reinstalling and need to make a backup of your components and protocols (xmldb)?	<ol style="list-style-type: none"> 1. Open a command prompt. 2. Navigate to <pps_install>/bin. 3. Run the following command: <code>DbUtil.exe -backup -file xmldb_backup.zip</code> <p>This will create the following file: <pps_install>/public/backups/xmldb_backup.zip.</p>	<input type="checkbox"/>

Installing on Windows

1. Review the [Pre-Installation Checklist for Windows](#). If you are upgrading, see [Chapter 4: Upgrading Pipeline Pilot](#).
2. Log on to the destination computer as an administrator.
3. Extract the BPP2021.7z file.
4. In the extracted folder, double-click `\bin\scitegicsetup.exe` to run the installer.
5. When the Pipeline Pilot installer starts, review the Welcome screen and then click **Next**.
6. If prompted, install the required Microsoft runtime libraries (DLLs).
7. At **Select a Destination Directory**, accept the default `C:\Program Files\BIOVIA\PPS` or click **Change** and navigate to the location where you want Pipeline Pilot installed, and then click **Next**.
8. Browse to or enter the path to your license file (*.lic). It determines what software and component collections to install on your server.

Note: The Pipeline Pilot licenses are made available when the software is purchased. For easy access, this license file should be copied to the server where you plan to install Pipeline Pilot. If you need further assistance relating to licenses you should initially contact your account manager.

9. At the **Select products to install** dialog, the products lists are collapsed. Expand the list to view which products are included with your Pipeline Pilot license. Click **Next** to install all the products. You can uncheck any product that you do not want installed. Review the list carefully.
By default, the installer backs up your XMLDB in `root/public/backups` (in compressed zip format).

Notes:

- The Platform product is installed by default.
- To install a collection update, click **Add/Remove Packages** after the installer starts. To reinstall the same version again, click **Reinstall/Repair**.

10. If the **WARNING! Running Applications** window appears, click **Stop Services** to close the services that appear in the dialog window.
11. Verify the port numbers for the Apache web server. The following are the default port numbers:

Type	Default Port
HTTP	9944
HTTPS	9943
Tomcat Shutdown	9945
Tomcat HTTP	9946
Jupyter	9947

If another service is already using these ports, you can assign alternate ports for HTTP and HTTPS during installation. Be sure to make a note of what you change so can easily refer to these ports

later. To reconfigure the other ports after installation, see the Pipeline Pilot Help Center: **Admin > Setup and Configuration > Server Configurations > Reconfiguring Ports**.

12. At **Confirm Your Installation**, click **Next** to start the installation.
13. If you want to view each package as it installs, check **Show installation progress in command window**.
14. At **Confirm Apache Login**, click **Next** if you do not want to modify the Apache user. To modify the Apache user, click **Apache** and edit the **Log On** property for your Apache service.
15. At **Installation Complete**, click **Finish** to exit the installer.
16. If you want to open the Pipeline Pilot Server Home Page, select the checkbox before you click **Finish**.
17. Continue with the [Post-Installation Steps](#).

Notes:

- If there is a problem during the installation, the last screen of the installer provides an option for viewing your installation log. It provides details about what took place during the installation, to help you pinpoint the nature of the problem. The file is saved in:
`<pps_install>\logs\install.`
- After Pipeline Pilot is installed, several services are started. See [Chapter 3: Pipeline Pilot Services](#). You can stop and restart these services from the Windows Services console.
- After Pipeline Pilot is installed, you have the option to install Pipeline Pilot Client and/or Data Source Builder on the server. The Pipeline Pilot Home Page also has an option for installing Pipeline Pilot Client and/or Data Source Builder (for details, see [Post-Installation Steps](#)). The default Pipeline Pilot Home Page URL is: `http://<server_name>:<port>` (for example, `http://myserver:9944`).
- Client users can install Pipeline Pilot Client by connecting to the Pipeline Pilot Home Page from their Web browsers, see the *Pipeline Pilot Client Installation Guide* for further information.

Windows Guidelines

Windows Requirements for Load Balancing

Refer to the *Pipeline Pilot Deployment Guide* and the [Chapter 8: Load Balancing Pipeline Pilot](#) section.

Local Job Temporary Directory

For best performance, it is possible to use a local disk for temporary files generated while a job is running. This minimizes network traffic and reduces job run time. The local job temp directory must exist on all nodes where the server is running in the exact same path. Because some protocols generate a great deal of temporary data, this directory should be located on a file system that is large enough to hold the data. For more information Jobs, see the *Pipeline Pilot Deployment Guide*.

Impersonation on Windows

For a Windows server to run client impersonation, the following is required:

- Clients and server must be part of the same Windows domain or trusted domains.
- For Pipeline Pilot user authentication, clients use a local or domain account. This allows the server to actually log into that authenticated user name.
- For full impersonation, configure the Apache service to run under the local system account. Alternately, you can configure the server to run under an account that has administrative privileges

on the server. The account must also be configured with the Windows Local Security Policy rights **Replace a process level token** and **Adjust memory quotas for a process**.

- For restricted impersonation, configure the Apache service to run under an account that has administrative privileges on the server. The account does not have to be a domain account, but should be configured with the Windows Local Security Policy rights **Replace a process level token** to allow access to network based resources.

Under impersonation, client users must have Read, Write, Execute, and Directory Browse access to the following subfolders in <pps_install> directory:

- xmldb
- web/jobs
- public/users
- public/data
- public/bin
- public/scripts
- bin (windows)

Pre-Installation Checklist for Linux

Review the checklist below to ensure that you perform all preliminary tasks before deploying the Pipeline Pilot Pipeline Pilot Server in a Linux distributed environment.

IMPORTANT! You must install BIOVIA Foundation Hub before installing Pipeline Pilot Server.

- If you are installing on Windows, see [Pre-Installation Checklist for Windows](#) and [Installing on Windows](#).
- If you are upgrading, see [Chapter 4: Upgrading Pipeline Pilot](#).

Do This	Details	Done
1. Obtain BPP2021.tgz.		<input type="checkbox"/>
2. Locate your BIOVIA license file (*.lic).	<p>A valid license file is current (not expired) and accessible during the installation. For each new license file that you obtain, instructions are available that explain how to update and activate the license file.</p> <p>Licenses are made available when the software is purchased. If you need further assistance relating to licenses you should initially contact your account manager.</p> <div> <p>Note: If you are planning to use the Materials Studio collection, you will need to perform additional installation steps after the initial installation. See Appendix D: Installing BIOVIA License Pack for Materials Studio Collection.</p> </div>	<input type="checkbox"/>
3. Configure a user and group account for installing Pipeline Pilot.	<p>This account should also be available on the cluster. The Pipeline Pilot user account does not require a login. Normal users should not be members of this group.</p> <ul style="list-style-type: none"> ■ This account group should only be used for this user. No other users should belong to this group. ■ This can be a no-login account. When deploying on a cluster or grid, the account must be accessible from all machines. 	<input type="checkbox"/>
4. To increase performance and minimize I/O bottlenecks in grid, cluster, or load balanced installations, increase the number of NFS daemons running on the primary Pipeline Pilot.	<p>Making this change requires restarting NFS:</p> <ol style="list-style-type: none"> 1. Increase RPCNFSDCOUNT from 8 to a minimum of 16. Uncomment and edit the RPCNFSDCOUNT property in <code>/etc/sysconfig/nfs</code> while logged in as root. 2. Restart the NFS service to enact the change: 	<input type="checkbox"/>

Do This	Details	Done
	<pre>systemctl restart nfs</pre>	
<p>5. Ensure that you have sufficient local disk space to install Pipeline Pilot software and related data. This also applies to the local job temporary directory. To use a directory other than /tmp, it is necessary to first set it up.</p>	<p>The primary Pipeline Pilot Server should be installed on a disk that is local to the machine that runs it. Installing on an NFS mounted disk might result in protocol failures and other issues.</p> <p>See Distributed File System and NFS Configuration.</p> <p>IMPORTANT! The file system mount point path must be the same on all nodes. If you install in /opt/BIOVIA, all cluster nodes need to access the Pipeline Pilot installation as /opt/BIOVIA. This also applies to load balanced configurations where each node has its own installation.</p>	<input type="checkbox"/>
<p>6. For load balanced installations, create the file system mount point. balanced configurations.</p>	<p>For example:</p> <pre># mkdir /opt/BIOVIA chown ppuser:ppgroup /opt/BIOVIA</pre>	<input type="checkbox"/>
<p>7. Export the file system mount point.</p>	<p>Edit the file /etc/exports and add entries for each node you want to export to. The no_root_squash option should be set.</p> <pre>/opt/BIOVIA/ Node1(rw,no_root_squash, sync) Node2(rw,no_root_squash, sync) #exportfs -a</pre> <p>See Distributed File System and NFS Configuration.</p>	<input type="checkbox"/>
<p>8. For each node in the cluster, mount the exported file system.</p>	<p>Edit the file /etc/fstab and add an entry for the exported file system using the following options:</p> <pre>pp_server:/opt/BIOVIA/opt/BIOVIA nfs rw,hard,intr,lock,noatime,acdirmin=1, acdirmax=1,acregmin=0,acregmax=0 # mount /opt/BIOVIA</pre> <p>See Distributed File System and NFS Configuration.</p>	<input type="checkbox"/>
<p>9. If you are installing on a grid, verify that you can submit a simple job to the grid engine and that it runs successfully.</p>	<p>Manually run a job through the system and verify that it works. This usually involves using the grid engine qsub command. For more information, see Post-Installation for a Grid or Cluster on page 57.</p>	<input type="checkbox"/>

Do This	Details	Done
10. If you are using R Statistics or Discovery Studio®, plan on installing these packages in a location that can be exported to the cluster with a single share.	For example, to install both R and Discovery Studio and export them with one mount, install Pipeline Pilot in /opt/BIOVIA/Pipeline Pilot, install R in /opt/BIOVIA/R, and install Discovery Studio in /opt/BIOVIA/ds. This allows you to export /opt/BIOVIA so the cluster can access and run all the applications. See the <i>R Software for Pipeline Pilot Server 2021 Installation and Configuration Guide</i> .	<input type="checkbox"/>
11. Certain system requirements might be required for the correct operation of Pipeline Pilot.	Before starting the installation, you can display the required system libraries by running the <code>check_system.sh</code> script located in the <code>bin/SystemChecker</code> folder in the installation media. See Appendix B: Required RPMs for a list of required libraries.	<input type="checkbox"/>
12. Are you installing on a machine with an older version or reinstalling and need to make a backup of your components and protocols (xmldb)?	<ol style="list-style-type: none"> 1. Open a command prompt. 2. Run the following command: <pre>. <pps_install>/linux_bin/ppvars.sh <pps_install>/linux_bin/Dbutil -backup -file xmldb_backup.zip.</pre> 	<input type="checkbox"/>

Notes:

- When installing Pipeline Pilot to run in a web farm configuration, install the software in the same location on each machine. (Pipeline Pilot is installed on every node and is not shared.)
- For a cluster or grid operation, Pipeline Pilot is installed once. The installation is then shared using the same path on all nodes.
- For more information on these deployment options, see the *Pipeline Pilot Deployment Guide*.

Installing on Linux

IMPORTANT! The installation scripts must include read, write, and execute privileges for the user who is installing.

1. Review [Pre-Installation Checklist for Linux](#). If you are upgrading, see [Chapter 4: Upgrading Pipeline Pilot](#).
2. Change the directory to the location of the installation files. This is the directory that contains the `sciinstall` script.
3. If using impersonation, create an account for this user. The account should use its own group. No regular users should belong to it.
4. Extract the contents from the `BPP2021.tgz` file:

```
tar -xvzvpf
```

5. If you are using Red Hat Enterprise Linux or later and have Security-Enhanced Linux enabled, see [Local Authentication and SELinux](#).
6. Run `sciinstall` as the non-root user you created in [Pre-Installation Checklist for Linux](#):

```
$ ./sciinstall
```

7. Enter **1** to install a new server, and enter a target directory for installing the Pipeline Pilot application. For disk space requirements, see the *Pipeline Pilot Server System Requirements* document.
8. Edit the path to your BIOVIA license file, including the license name. Your license determines what software and product collections to install on the server.

Note: The Pipeline Pilot licenses are made available when the software is purchased. For easy access, this license file should be copied to the server where you plan to install Pipeline Pilot. If you need further assistance relating to licenses you should initially contact your account manager.

9. The installer displays the *Products* that will be installed, the disk space required for the installation, and the available space. Enter **y** to install Pipeline Pilot and all the available collections.
The *Platform* components are installed by default.
If you only want to install Pipeline Pilot or specific collections, enter the number of what you do not want installed.

Note: The *Action* listing for these products has changed to Read Remove or Embed. The embed collection packages will not be removed because they are required for a collection that you want to install.

When you have finished your selections, enter **y**.

To install a collection or a collection update, to reinstall an existing installation, or to upgrade an existing installation, rerun the installer and enter a number (1-5) for the action you want to complete.

10. Assign the HTTP and HTTPS ports (required by the Apache web server). The default ports are 9944 (HTTP) and 9943 (HTTPS). Enter **y** to accept the default port numbers
To assign alternate port numbers, enter **n**, and then enter the alternate numbers as the prompt.

Tip: The installer checks the ports to ensure that they are available. If it detects a conflict, a warning is displayed and the installation stops. To assign different port numbers, ensure that they are available before you rerun the installation program.

11. Select the local directory for lock files (that is a directory that is not mounted using NFS). The default directory is `/var/tmp`. The lock file directory requires, read, write, and execute privileges for all users. For cluster configurations, this directory path must exist on the local hard drive of each node.
12. Enter an option number when asked if the machine you are installing on is the primary node in a cluster. The options are:
 - This node is not in a cluster.
 - Round Robin.
 - Job Leveling.
13. Continue with the [Post-Installation Steps](#) and [Post-Install Checklist for Linux](#).

Notes:

- Depending on your installation choices, it might take some time for the installation to finish. When complete, the Apache and Pipeline Pilot Servers both start automatically.
- If you did not configure autostart, you need to manually start the server.

Linux Guidelines

Installation paths

Regardless of your environment, when installing Pipeline Pilot in a multiple node configuration, the installation path must be the same on all nodes that access Pipeline Pilot software. For example, if you install the software in `/opt/BIOVIA` on one machine, the software must be available in `/opt/BIOVIA` on all machines. This includes installations for load balanced configurations and shared installations such as grids and clusters. For more information on these deployment options, see the *Pipeline Pilot Deployment Guide*.

Red Hat Linux

Tip: Red Hat no longer ships with Xvfb, which is used by protocols that use R to generate images. RPM for Xvfb is available for download through Red Hat support (see <https://www.redhat.com/>).

Local Authentication and SELinux

If you are using local authentication or planning to use Security-Enhanced Linux (SELinux), it is necessary to allow local password file authentication.

Change the permissions on `/etc/shadow` so it can be read:

```
chgrp ppgroup /etc/shadow
chmod g+r /etc/shadow
```

Tip: This task is only necessary if you plan on using local accounts.

Distributed File System and NFS Configuration

For a clustered or grid environment, the primary server installation must be exported to the other nodes in the cluster. For load balanced configurations, the installation is not shared, but a shared file system is required for users, jobs, and public directories.

Tip: You might get better performance if you use a file system designed for clustering, such as GPFS, Panasas® or Lustre™.

The primary requirement for Pipeline Pilot is support for file locking and minimizing directory attribute caching. If you plan on using impersonation with Pipeline Pilot, it is necessary to turn off root squash for NFS mounts.

It is important when dealing with distributed file systems to keep your server clocks synchronized and ensure that the Network Time Protocol daemon (ntpd) is running on all nodes.

NFS Settings

Setting	Description
rw	Read and write enabled
hard	Hard mount option, required by mount. The alternative is soft, which does not work with the Pipeline Pilot.
intr	System calls can be interrupted. Without this option, some operations can cause the server or jobs to lock up while waiting on an NFS system call.
lock	This is required for the Pipeline Pilot Server. All file operations are controlled by advisory locking. If locking is disabled on the mount, the Pipeline Pilot Server will not work. In some cases where this was omitted for a mount, the server appears to work normally, because lockd is already running on the machine. If there is an existing lockd running, it will handle the lock calls and things will appear to work normally. However, the Pipeline Pilot Server might stop working correctly if the server is rebooted.

NFS Cache Settings

Setting	Description
acdirmin, acdirmax	These settings control attribute caching for directories. Directory attribute caching determines the amount of time it takes for a file to show up on other machines. Values specified are in seconds. If directory attribute caching is enabled, operations that depend on a recently written file existing will fail intermittently. The required settings for these are <code>acdirmin=0</code> , <code>acdirmax=0</code> .
acregmin, acregmax	These settings control file attribute caching. File attribute caching determines the amount of time it takes for file ownership and permission changes to show up on other machines. Values are specified in seconds. The recommended settings for these are <code>acregmin=0</code> , <code>acregmax=0</code> . Note: <code>acregmax</code> should not be set to more than 10 seconds.
rsize, wsize	These are the read and write transfer sizes. System defaults should be used. To improve performance, these settings can be safely increased.
noatime	Disables setting the access time on the file. This is a performance optimization and should be used on all remote mounts.

Tip: A good starting point for specifying NFS mount options for Pipeline Pilot Server is: `rw,hard,intr,lock,noatime,acdirmin=0,acdirmax=0,acregmin=0,acregmax=0`

Clock Synchronization

In a distributed environment, it is important for all nodes to have synchronized clocks. Ensure that the Network Time Protocol daemon (ntpd) is running on all nodes in the cluster, and that all nodes report the same time.

Fixing Java JDK Timezone Issues

In some cases, Pipeline Pilot code that executes within the Java JDK (such as the *Java (on server) component*) incorrectly detects the timezone setting of the operating system. You can address this limitation within Java setting the TZ environment variable:

1. Open a bash shell:

```
sudo bash
```

2. Edit the environment:

```
vi /etc/environment
```

3. Add the following text:

```
TZ=America/Los_Angeles (Use the timezone for your location)
```

4. Save.

Possible Interaction with /usr/sbin/tmpwatch

If the server is idle for a period of time, it is possible that the tmpwatch directory cleaner can end up deleting the /var/tmp/scitegic-mutexes-XXXX/.lock file. Under impersonation, this can result in the lock file ending up with the wrong permissions and causing problems on the system.

If you are running tmpwatch, edit its configuration file to exclude the /var/tmp/scitegic-mutexes-XXXX directory.

Chapter 6:

Post-Installation

Post-Installation Steps

After installing Pipeline Pilot on your server, there might be additional configuration steps required for connectivity and resource access. BIOVIA recommends that you review the information provided below and identify any outstanding tasks you might need to complete.

For Linux, see also the [Post-Install Checklist for Linux](#).

Note: There are many additional tasks you can perform on the server in addition to the ones listed below. For more detailed information, see the Pipeline Pilot Admin Portal help.

Would you like to:	Here is how:
Set up users, groups, and permissions? You can use Pipeline Pilot to authenticate users, or if you are installing Pipeline Pilot as part of a suite of BIOVIA products, you will use Foundation Hub to manage users, groups, and permissions.	See the <i>Foundation Hub Installation and Configuration Guide</i> . Note: If you have custom groups and assignment and you are installing with Foundation Hub for the first time, you should run the <i>Hub User Synchronization from Active Directory</i> protocol to migrate your custom authentication data. See Migrating custom Pipeline Pilot authentication data to Foundation Hub .
Get more familiar with Pipeline Pilot? Visit the Pipeline Pilot Server Home Page and launch the Pipeline Pilot Admin Portal. Install the Pipeline Pilot Client and run example protocols.	From your server's web browser, go to: <code>http://<servername>:9944</code> Where <servername> is the name of your server.
Learn about managing Pipeline Pilot? Review the online help available in the Pipeline Pilot Admin Portal, and get more familiar with the various ways you can configure your server.	From the Pipeline Pilot Server Home Page, select Help Center > Administrator Resources .
Verify that your collections can properly run? Review the documents included with each separately installed collection to ensure that all requirements are met. In some instances, third-party software might be required on the server or client machines to support a collection.	From the Pipeline Pilot Server Home Page, select Help Center > User Resources . A set of documents that explains how to support and use each collection is available on a per-collection basis (organized in alphabetical order).
Share a protocol database across servers? Clients running at multiple sites can access the same protocols and components if they share the same XMLDB .	Change the XMLDB Endpoint in the Pipeline Pilot Admin Portal so that it points to the server that hosts the protocol database (Setup > Server Configuration).
Authenticate users on the server?	Configure authentication in the Pipeline Pilot

Would you like to:	Here is how:
You can restrict access to the server based on the identity of your network users.	Admin Portal (Security > Authentication).
Enable impersonation? Impersonation allows users to access network resources using their network security credentials rather than their server account credentials.	Enable impersonation in the Pipeline Pilot Admin Portal (Security > Authentication).
Restrict publishing rights on the server? All users have read/write access to the shared tabs, allowing all users the rights to “publish”. You can control how users publish protocols on these shared tabs by managing access rights.	Configure access rights in the Pipeline Pilot Admin Portal (XMLDB > Access Rights).
Support Web Port users? Any client user, who needs to run protocols remotely or from a web browser, needs to run Web Port.	A link is available from the Pipeline Pilot Server Home Page for launching Web Port.
Provide database connectivity? ODBC drivers from DataDirect are included with the Integration collection. These drivers support connections to Oracle, Microsoft SQLServer, MySQL, and DB2. A version of the Oracle JDBC driver for connecting to Oracle databases is also available. To use the Integration components with your database, you need to configure data sources on your server.	Read the instructions in the <i>Database Integration Guide</i> in the Pipeline Pilot Help Center on the Developers tab. ODBC Driver Help is also available on the Pipeline Pilot Server Home Page.
Schedule protocol jobs? You can run protocols outside of the graphical client environment on any computer where Pipeline Pilot is installed with job scheduling.	Install the Integration collection and run command-line program <code>RunProtocol.exe</code> . Read the <i>Command Line Guide</i> in the Pipeline Pilot Center on the Developers tab.
Migrate an existing Pipeline Pilot installation to a new machine? Reuse your protocol database (XMLDB) in the new environment.	Point all clients (Pipeline Pilot Client and web-based) to the new host, and install the required software on the new host. Update your browser bookmarks to Web Port and any web pages that contain links to protocols. For further details, see Moving or Migrating an Installation .
Upgrade an existing installation? Depending on your installed packages, some additional steps might be necessary to get your server properly configured and running.	Activate any non- Pipeline Pilot packages, such as custom Java components or third-party components, and recompile your client applications that use the Java SDK or .NET SDK with the new versions of <code>jalpp.jar</code> or <code>nalpp.dll</code> .
Upgrade an existing installation and modify the port numbers? See the <i>Pipeline Pilot Administration Guide</i> section	Update your browser bookmarks to Web Port and any web pages that contain links to protocols.

Would you like to:	Here is how:
on Reconfiguring Ports. Also, any applications that rely on the new port numbers will need to be updated.	
Deploy Pipeline Pilot Server on a grid, cluster, web farm, or reverse proxy? Ensure your machines are properly configured to support these features.	See the <i>Pipeline Pilot Deployment Guide</i> .
Configure the Java Server package that you installed as part of the server installation? Some additional steps might be necessary.	Before you can start the Java Server services, validate the ports that you need to use in the Admin Portal (Processes > Server Ports). After validation, you can start the Java Server services from the Pipeline Pilot Admin Portal.
Add lines to your <code>.bash_profile</code> or <code>.cshrc</code> file to source either <code><pps_install>/linux_bin/ppvars.sh</code> or <code>ppvars.csh</code> .	For the C-Shell, add the following line to <code>~/.cshrc</code> <code>source <pps_install>/linux_bin/ppvars.csh</code> For the Bash shell, add the following line to <code>~/.bash_profile</code> <code><pps_install>/linux_bin/ppvars.sh</code> Where <code>~</code> in the path resolves to your home directory on the Linux server.

Post-Install Checklist for Linux

After installing Pipeline Pilot, you can configure features in the Pipeline Pilot Admin Portal to support a particular deployment method. On Linux it is necessary to perform a few preliminary tasks depending on your operating system and planned deployment.

Do This	Details	Done
1. Shut down the Pipeline Pilot Server if it was started during the installation process.	For example: <pre>cd /opt/BIOVIA/installdir/linux_bin ./stopserver</pre>	<input type="checkbox"/>
2. Run <code>scirootinstall</code> .	<code>scirootinstall</code> is located in the <code><pps_install>/apps/scitegic/installer/scripts</code> directory. Even if you do not intend to use impersonation, it's a good idea to run <code>scirootinstall</code> at this time. This script sets up directory permissions, configures autostart, sets up PAM, and configures the server to run with impersonation. Note: If you are setting up a cluster, <code>scirootinstall</code> should be run on all nodes in the cluster.	<input type="checkbox"/>
3. Set group directory access. This applies when using alternative locations for the users, jobs, or the XMLDB. This needs to be handled so your server can access files created by impersonated users.	Since <code>scirootinstall</code> does this for the default installation, this step is only necessary if you use an alternate location. <pre>mkdir AltUser chown ppuser:ppgroup AltUser chmod 2775 AltUser</pre>	<input type="checkbox"/>
4. Run <code>mkpublic</code> . This is required when using a shared public directory in load balanced configurations.	<code>mkpublic</code> is located in the <code><pps_install>/apps/scitegic/installer/scripts</code> directory. It can be run either as root or the server user. This command creates a compatible public directory with the correct permissions. For example: <pre>mkpublic [-u user][-g group] path</pre>	<input type="checkbox"/>

Do This	Details	Done
5. For a grid, run <code>scigridsetup</code> on each node that will run protocols.	<p><code>scigridsetup</code> is located in the <code><pps_install>/apps/scitegic/installer/scripts</code> directory. This script adds an <code>init</code> script for Xvfb and runs <code>sciseallow</code>.</p> <p>IMPORTANT! If you do not run Xvfb, components that use R software to generate plots and graphics might not work. For further details, see the <i>R Software for Pipeline Pilot Server 2021 Installation and Configuration Guide</i>.</p>	<input type="checkbox"/>

Testing Client Software on Linux

The Pipeline Pilot Client *cannot* be installed on a Linux server. To verify the server installation, you must either connect to the server through a Pipeline Pilot Client on a separate Windows machine or use a web browser to access the Pipeline Pilot Server Home Page.

To access the home page:

- Go to `http://<servername>:<port>`

Where <servername> is the name of your Linux server and <port> is the port used for the Pipeline PilotService.

To:	Select:
Read documentation about using and administrating Pipeline Pilot	Help Center
Install Pipeline Pilot Client on Windows	Install/Upgrade Client Software
Change server settings or enable impersonation	Administration Portal
Launch the Pipeline Pilot web client	Web Port

Server Configuration Checklist

After installing Pipeline Pilot, you can configure features in the Pipeline Pilot Admin Portal to support a particular deployment method. Review the following checklists before you make any changes in the Pipeline Pilot Admin Portal.

Do this	Details and Help Location (bold)	Done
1. Restart the Linux server if you previously stopped it to perform any tasks.	<code>cd /opt/BIOVIA/installldir/linux_bin ./startserver</code>	<input type="checkbox"/>
2. Log into the Pipeline Pilot Admin Portal and configure the options. If you have installed Pipeline Pilot as part of a suite of applications, you will use Foundation Hub to manage users, groups, and permissions.	See: Security > Authentication	<input type="checkbox"/>
3. If you need to use impersonation, enable it first, and then verify that it works before making other configuration changes.	Enable impersonation in the Admin Portal. See: Security > Authentication.	<input type="checkbox"/>
4. For clustering support (Linux only), log in to each of the cluster nodes and start the servers. Add the nodes to the cluster node list and verify that the nodes are active (green). Round robin is the recommended clustering method.	<code>ssh node1 cd /opt/BIOVIA/installldir/linux_bin ./startserver</code>	<input type="checkbox"/>


Do this	Details and Help Location (bold)	Done
5. If you are configuring for a load balancer, export your configuration and save the exported file to make it easy to deploy new nodes with the same configuration.	Export and import your server configuration in the Admin Portal. See: Maintenance > Export Import Configuration See also load balancing Pipeline Pilot Server Administration Configuration .	<input type="checkbox"/>

Pipeline Pilot Server Home Page

The Pipeline Pilot Home Page is a one-stop place for all things related to Pipeline Pilot, Pipeline Pilot Client, and other BIOVIA applications. It opens in a browser window and provides a way to get assistance and access to documentation and programs. From here, you can:

- Install or upgrade Pipeline Pilot Client software.
- Access documentation in the Pipeline Pilot Help Center.
- Start the Pipeline Pilot Web Port.


Administrators can perform server tasks in the Pipeline Pilot Admin Portal and install Data Source Builder and BIOVIA Draw.




Server Home Page

Pipeline Pilot


Server: pps-win Version: 18.1

 **Pipeline Pilot Users**


- [Help Center \(Client Users\)](#)
- [Install/Upgrade Client Software](#)
- [Client Installation Guide](#)

 **Administrators**


- [Help Center \(Administrators\)](#)
- [Administration Portal](#)
- [DataDirect Driver Help](#)

 **Developers**

- [Help Center \(Developers\)](#)
- [Web API Reference](#)

 **Web Port**

- [Web Port](#)

 **RSS Feeds**

- [RSS Server Feeds](#)

 **RSS Feeds**

- [RSS Server Feeds](#)

Technical Support

If you have any questions, contact BIOVIA Customer Support.

Support for BIOVIA Products:

- biovia.support@3ds.com
- biovia.jp.support@3ds.com

BIOVIA Support Offices:

A complete list of regional BIOVIA Customer Support offices is available at:

- <http://accelrys.com/customer-support/contact.html>

Application Links

- [Install BIOVIA Draw](#)
- [Install Data Source Builder](#)
- [Install BIOVIA Plugin](#)

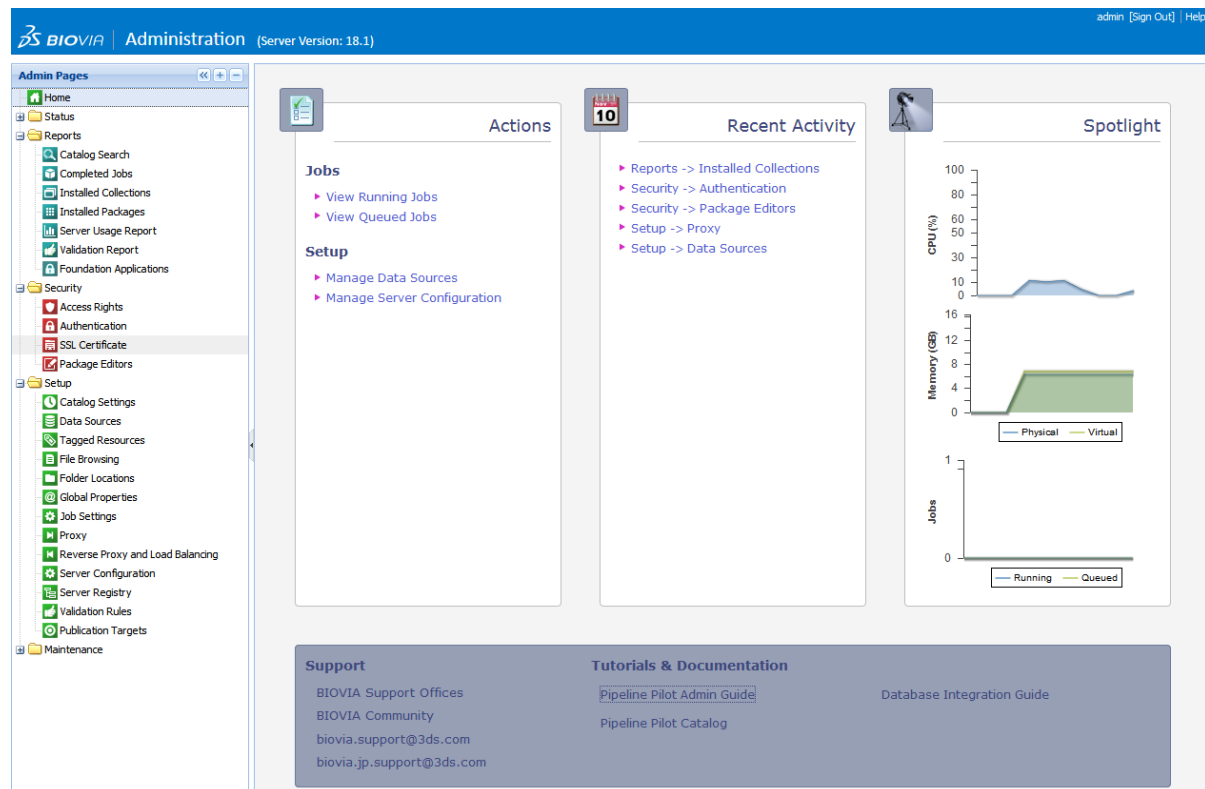
Pipeline Pilot Administration Portal

System administration and server configuration tools are available in the Pipeline Pilot Administration Portal (Admin Portal). This web application provides secure access from any network location, independent of the server location. It runs with Internet Explorer and Firefox or another Mozilla-supported browser.

Tip: To access the Admin Portal, you need a valid user name and password combination. The default settings are: "scitegicadmin" (user name) and "scitegic" (password). The first time you log into the Admin Portal, you can use these credentials.

To log into the Admin Portal:

1. From the Pipeline Pilot Server Home Page, select **Administration Portal**. The login page opens.
2. Enter the default user name and password and click **Log In**.



Pipeline Pilot Admin Portal - Home page

Note: See the Pipeline Pilot Admin Portal online help for further details on using this application, click **Help** in the top right corner.

Installing Pipeline Pilot Client

1. If you have an existing installation of Pipeline Pilot Client, uninstall it before installing the new version.
2. In your web browser, go to the Pipeline Pilot Server Home Page:
`http://<server_name>:<port>`
where <server_name> is the host name of your server and <port> is the port number (9944 by default).
3. Select **Install/Upgrade Client Software**. The Pipeline Pilot Client installation page opens.

Install Pipeline Pilot
To install the client software:

1. Make a note of your server's host name and port number:
`https://aep9-webapps:9943`
2. Download and run the [Client Installer](#).
3. After the installation is complete, use the host name and port number listed above to start Pipeline Pilot and connect your client to the server.

4. Copy the name and port number of the server currently used by your browser for the home page connection. You will connect to this same server after you complete the installation.
5. Click **Client Installer** to download `clientsetup.exe`.
6. Run `clientsetup.exe`.
7. When the installer starts, review the welcome and end user license agreement screens and follow the prompts to install.
8. Select the location where you want to install the Pipeline Pilot Client. The default location for Pipeline Pilot Client is: `C:\Program Files\BIOVIA\Pipeline Pilot Client <version>`
9. Click **Install**. When the installation is complete, the installer screen provides you with the option to start Pipeline Pilot Client automatically. You might be prompted to specify the server name and port or log in using a valid username and password when you start Pipeline Pilot Client.

Tip: To ensure that all required software is installed, run example protocols using the default settings. For assistance, contact your system administrator or contact [Dassault Systèmes Customer Support](#).

Setting Foundation Hub as the Authentication Server

You can use Foundation Hub to manage Pipeline Pilot authentication.

If you are using your Pipeline Pilot Server to deploy other BIOVIA applications or integrate with other BIOVIA products you *must* employ Foundation Hub authentication.

Refer to the *Pipeline Pilot Admin Portal Help* for more information on other authentication options.

1. Open the Pipeline Pilot Administration Portal.
2. Open the **Security > Authentication** page.
3. Set **Check users against** to **An external user directory**, and select **Foundation**.
4. Set **Foundation Server URL** to the fully qualified domain name of a Foundation Hub server (or a load balanced endpoint). For example:
`https://<hubserver>.com:9953/foundation/hub`
 Or:
`https://<hub_lb_server>.com:9953/foundation/hub`
5. Enter credentials for the **Foundation Admin**: `scitegicadmin/scitegic` by default.
6. Click **Save**. After a few moments, the page will indicate a successful registration.

Refer to the *Foundation Hub Administration Guide* for more information about managing security after you have set the Foundation Hub to manage Pipeline Pilot authentication.

Note: There may be a logout delay for Pipeline Pilot of up to 60 seconds if the user logs out from Foundation Hub.

Migrating Custom Pipeline Pilot Authentication Data to Foundation Hub

If you have custom groups and assignment and are installing with Foundation Hub for the first time, you should run the *Hub User Synchronization from Active Directory* protocol to migrate your custom authentication data.

Notes:

- Custom Pipeline Pilot groups are imported.
- Custom Pipeline Pilot permissions *are not* imported.
- Assignments to groups and permission are imported as follows:
 - Group assignments: User and subgroups assignments are imported.
 - Permission assignments: Group assignments are imported. Since a user cannot be directly assigned to a permission in the Foundation Hub data model, a group named `<username> user permissions` is created and this group is assigned to the permissions.

To run the protocol:

1. Login to the Pipeline Pilot Server as a user with Foundation Hub administration privileges (`scitegicadmin` has this permission).
2. From the Pipeline Pilot Server Home Page, navigate to **Web Port > Foundation Hub Integration** and open **Hub Authorization Data Import from Pipeline Pilot**.
3. Select the source authorization data file.
 This is an XML file automatically generated when Pipeline Pilot Server is connected to Foundation Hub.
4. Check the following checkboxes:
 - **Simulate import to Foundation Hub:** Run a test without actually importing the data.
 - **Rollback on Foundation Hub error:** Do not import the data if an error is encountered. Warnings are ignored.
 - **Import externally managed Users:** Create users that are not defined in the Pipeline Pilot database, but are assigned to existing groups or permissions (for example, users represented

by an external identity provider.)

Synchronizing with a user directory instead can help you clean out assignments to users that no longer exist. Users created by this option are only defined by their username (information such as name or mail address are not imported). If you have domain users assigned to groups and permissions, it is recommended that you run the *Hub User Synchronization from Active Directory* protocol before importing the Pipeline Pilot data.

- **Import Pipeline Pilot File Users:** Import users that are assigned to groups and permissions.

Notes:

- Passwords are not exported from Pipeline Pilot. You must reset the passwords in Foundation Hub before users can sign in.
- Groups and permissions that do not have users are not imported.
- Run the protocol in simulate mode to view the data that would be imported before committing it to the Foundation Hub.

5. Click **Next>>** to run the protocol.

Chapter 7:

Uninstalling Pipeline Pilot

Before Uninstalling

Before uninstalling software on any Pipeline Pilot Server, do the following (in order):

1. In the Pipeline Pilot Admin Portal, open the **Status > Running Jobs** page.
Confirm there are no jobs running on the server.
2. Back up the XMLDB:
 - a. Open the **Maintenance > Server Backup** page.
 - b. In the **Setup Backup Folder** section, confirm the path where the backup archive will be stored.
 - c. Check that the **Backup file name** is appropriate and change it if necessary.
 - d. Click **Create Backup**. This creates a file based on the date and save it to your backup path.
3. Open the **Setup > Folder Locations** page. If you customized the location of your **XMLDB Directory**, reconfigure it to the <pps_install>/xmlldb default location.
4. Open the **Security > Authentication** page and disconnect from your Foundation Hub server. Either change the external directory or turn off checking users against an external directory.
5. Shut down the [Pipeline Pilot Services](#).

Uninstalling on Windows

Use the Windows Services console to shut down all services that are currently using the ports for the software you need to remove. For more information, see [Pipeline Pilot Services](#).

You can also stop all Pipeline Pilot services when you start the Pipeline Pilot installer, but you still have to manually shut down Apache for earlier releases.

1. Open the Windows Services console.
2. Shut down **BIOVIA Pipeline Pilot 21.1.0 (Manager)**.

Note: Shutting down this service will also shut down the following sub-services: BIOVIA Pipeline Pilot 21.1.0 Service (Httpd)
BIOVIA Pipeline Pilot 21.1.0 Service (Tomcat).

3. Use **Add/Remove Programs** in the **Windows Control Panel** to uninstall Pipeline Pilot Server.
For Windows 8.1 or above, right-click the Pipeline Pilot icon in the Windows Start Menu and select **Uninstall**.

Tip: When you uninstall Pipeline Pilot, your protocol database (XMLDB) remains on your server. You can install a new version in the same location. You can also migrate your XMLDB to a new server if you plan to reuse it somewhere else. See [Moving or Migrating an Installation](#).

Uninstalling from a Linux Server

1. Run `ppvars.sh` before stopping the server.
2. Stop the server using the `stopserver` command in the `/opt/BIOVIA/installdir/linux_bin` directory.
`./stopserver`
3. Disable boot startup and shutdown:
 - If you are using a newer operating system that uses `systemd` for autostarting:
 - a. Execute the following command as root:
`systemctl disable ppilotd.<port>`
 - b. Remove the following file: `/etc/systemd/systemd/ppilotd.<port>.service`.
 - If you are using an older operating system that does not use `systemd` for autostarting:
`rm /etc/rc.d/rc*.d/*ppilotd. <port>`
`rm /etc/init.d/ppilotd. <port>`
4. If you installed on a cluster or a grid, perform these steps for each node:
 - a. Stop the server (cluster only):
`/opt/BIOVIA/installdir/linux_bin/stopserver`
 - b. Unmount the exported file system:
`umount -f /opt/BIOVIA`
 - c. Edit `/etc/fstab` and remove the entry for the exported file system.
 - d. Remove the `scitegic mutex` directory (cluster only):
`rm -rf /var/tmp/scitegic-mutexes-<port>`
 where `<port>` is the server port number
 - e. Remove the `/etc/pam.d/scitegic` file (cluster only):
`rm /etc/pam.d/scitegic`
 - f. If you installed the boot scripts:
 - If you are using a newer operating system that uses `systemd` for autostarting:
 - a. Execute the following command as root:
`systemctl disable ppilotd.<port>`
 - b. Remove the following file: `/etc/systemd/systemd/ppilotd.<port>.service`.
 - If you are using an older operating system that does not use `systemd` for autostarting:
`rm /etc/rc.d/rc*.d/*ppilotd. <port>`
`rm /etc/init.d/ppilotd. <port>`
 - g. If you installed the SELinux module, remove it:
`/usr/sbin/semodule -r scitegic`
 - h. If you installed the `Xvfb init` script (grid only), remove it:
 - `rm /etc/rc.d/rc*.d/*scixvfb.<port>`
 - `rm /etc/init.d/scixvfb.<port>`
 where `<port>` is the server port number

5. If you installed on a cluster, remove the export entries.
Edit `/etc/exports` and remove the entries for the exported file systems.
6. Remove the scitegic mutex directory:
`rm -rf /var/tmp/scitegic-mutexes-<port>`
where `<port>` is the server port number.
7. Remove the installation:
`rm -rf /opt/BIOVIA/installdir`
8. Remove the user and group entries for the Pipeline Pilot Server:
`userdel -r <username>`
where `<username>` is the name of the Apache user.
`groupdel <groupname>`
where `<groupname>` is the name of the Apache group.
9. Remove the `/etc/pam.d/scitegic` file:
`rm /etc/pam.d/scitegic`
10. If you installed the SELinux module, remove it:
`/usr/sbin/semodule -r scitegic`

Chapter 8:

Load Balancing Pipeline Pilot

Load Balancing Server Requirements

A load balanced or web farm configuration consists of a load balancer and multiple servers. The load balancer is a virtual server shared between all other servers, it acts as a reverse proxy and distributes network or application traffic across a number of servers. Load balancers are used to increase capacity (concurrent users) and reliability of applications.

Before you decide to set up a load balanced configuration you should review your requirements. Refer to the *Pipeline Pilot Server Deployment Guide* for guidance when assessing your needs.

You should also review the *Pipeline Pilot Admin Portal Guide* sections on **Configuring a Reverse Proxy** and **Load Balancing Deployments**.

To prepare a load balanced configuration you require:

- The fully qualified domain name and IP address of the load balancer Pipeline Pilot Server, for example:

```
<lb_server>.mycompany.com  
192.168.2.100
```

Where *<lb_server>* is the hostname of the load balancer Pipeline Pilot Server.

- The server names for each server. For example:

```
<server1>.mycompany.com  
<server2>.mycompany.com  
<server3>.mycompany.com  
...
```

- The same ports should be available and open on the load balancing Pipeline Pilot Server and all other Pipeline Pilot Servers.

The default Pipeline Pilot ports are 9944 and 9943.

This ensures that a load balancing server address redirects to the other servers properly, for example:

```
https://<lb_server>.mycompany.com:9943/admin
```

Tip: These example server names will be used throughout this document to demonstrate configuration steps.

Application Specific Customizations

If you are going to use your load balanced Pipeline Pilot Servers to run another application you should must ensure that all servers are identical in every respect, including:

- Package settings and global properties
- Manual package configuration or customization
- Customized protocols or components

If you have customized protocols or components these must be identical across all Pipeline Pilot Servers in the load balanced configuration. To achieve this:

1. Create a new deployment package on a test Pipeline Pilot Server (where you have validated all customizations).
2. Copy the deployment package to all load balanced servers.
3. Install the deployment package.

For more information on creating packages and zipping them for distribution refer to the *Application Packaging Guide* on the Developers tab of the Pipeline Pilot Help Center.

For example, there are additional configuration steps for Biological Registration and Chemical Registration that must be performed at the correct point in load balancing configuration, refer to the *BIOVIA Biological Registration Installation Guide* or *BIOVIA Chemical Registration Installation Guide* for more information.

IMPORTANT! You must *not* deploy any components or protocols in your XMLDB (that is, locally saved) on any or all Pipeline Pilot Servers in a load balanced configuration. All components and protocols *must* be installed using Pipeline Pilot packages.

Preparing Servers for Load Balancing

Pipeline Pilot Servers that will be used in a load balancing configuration must be prepared to ensure that they all have equivalent configurations that the load balancer can interact with. All Pipeline Pilot Servers including the load balancer need to have access to some shared folders.

Creating Required Folders

Before configuring your Pipeline Pilot Servers you must create three network folder locations for use with all load balanced Pipeline Pilot servers.

- A **User** directory to hold user folders. Each user has a folder for writing files for long-term storage. For example:
\\networkshares\shareduser
- A **Jobs** directory stores temporary files created by protocol jobs. It serves as current directory for job execution and holds temporary files while a job is running and job result files for the job's lifetime. For example:
\\networkshares\sharedjob

IMPORTANT! If you are going to use your load balanced Pipeline Pilot Servers to run BIOVIA Biological Registration or BIOVIA Chemical Registration you must ensure that the jobs folder path ends \web\jobs in order for all registration processes to function properly.

- A **Shared Public Directory** allows shared files to be accessed when Pipeline Pilot Servers are running a load balanced configuration. For example:
\\networkshares\sharedpublic

Note: It may also be useful to create a network folder to transfer configuration files between your Pipeline Pilot Servers, for example:

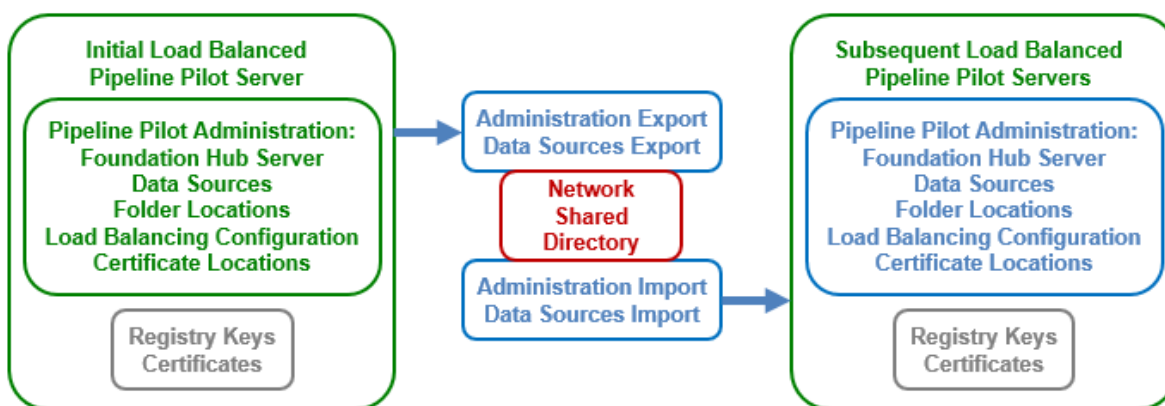
\\networkshares\loadbalancingconfig

Tip: For Linux load balancing configurations, refer to the [Distributed file system and NFS configuration](#) section when preparing your shared folders.

General Approach

When configuring a set of Pipeline Pilot Servers for load balancing the following approach is recommended.

- Configure a **primary** Pipeline Pilot Server for load balancing.
- Configure registry keys and certificates on the **primary** Pipeline Pilot Server.
- Export the **primary** Pipeline Pilot Administration and Data Sources configuration (for example, to a network shared directory so that the export files are accessible to all other Pipeline Pilot Servers).
- Import the **primary** configuration points to all load balancing Pipeline Pilot Servers.
- Configure registry keys and certificates on all load balancing Pipeline Pilot Servers.



Selecting Pipeline Pilot Administration Settings

Before you configure your Pipeline Pilot Servers for a load balancing configuration you should decide on some settings to use in order to maximize the benefits of employing load balancing.

The Pipeline Pilot Administration Portal can be used to configure the **Maximum Number of Simultaneous Parallel Processing Subprotocol Jobs** (on the Setup > Job Settings page).

In general, use the default value (4) or less to provide more even load balancing across users.

If your load balanced configuration is being used for a web-based application set this to 0 or 1.

Setting the Pipeline Pilot Service User (Windows)

Perform the following steps on *each* Pipeline Pilot Server (not including the load balancer server).

1. Open the Windows Services management console.
2. Right-click on **BIOVIA Pipeline Pilot 21.1.0 (Manager)** and click **Properties**.
3. On the **Log On** tab select the **This account** radio button.
4. Enter the username and password for the user account that you want to employ.
5. Click **OK**.
6. Repeat these steps using the same settings for the other Pipeline Pilot services:
 - **BIOVIA Pipeline Pilot 21.1.0 Service (Httpd)**
 - **BIOVIA Pipeline Pilot 21.1.0 Service (Tomcat)**

Note: The user account must have the following properties:

- A domain account on each Pipeline Pilot Server.
- Membership of the **Administrators** Group on each Pipeline Pilot Server.
- **Modify, Read, Write, Read & Execute**, and **List Folder Contents** permissions on all of the network shared folders.

Tip: To verify that each Pipeline Pilot Server machine can contact the load balancer server, open the command prompt and use the ping command:

```
ping <lb_server>.mycompany.com
```

Configuring a Primary Pipeline Pilot Server for Load Balancing

First, configure a primary Pipeline Pilot Server ready for load balancing. You will export this configuration and then import it onto other Pipeline Pilot Servers to ensure that they all have identical configurations.

Tip: You should perform all the primary Pipeline Pilot configurations on the same server so that when you export the Pipeline Pilot Administration configuration all the settings are included. If you configure Certificate information on one Pipeline Pilot Server and set the load balancing proxy information on another you will not be able to export all the configuration information together.

Setting up the Registry Keys (Windows)

1. In Windows, open the **Registry Editor**.
 - a. Navigate to **HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > services > LanmanWorkstation > Parameters**.
 - b. Right-click **Parameters** and select **Export**.
 - c. Export this registry entry as a back up for subsequent changes.
2. In Windows Explorer, navigate to the <pps_install>\install\templates\win_noarch folder.
3. Double-click on LoadBalancedConfig_Enable.reg. This will add the following information to the Windows Registry:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Lanmanworkstation\Parameters]
"FileInfoCacheLifetime"=dword:00000000
"FileNotFoundCacheLifetime"=dword:00000000
"DirectoryCacheLifetime"=dword:00000000
```

Notes:

If you are installing on load balanced Windows 2012 R2 Servers and have updated VMWare Tools and applied Microsoft Updates, you may encounter an issue where the Pipeline Pilot Server Administration Portal cannot be opened and protocols cannot be run. This issue is likely caused by the removal of needed registry entries during the update. To fix this re-run LoadBalancedConfig_Enable.reg as described above in step 3 to re-add them.

Tip: If you need to remove a Pipeline Pilot Server from load balancing you can either reinstate the exported registry settings or run the <pps_install>\install\templates\win_noarch\LoadBalancingConfig_Disable.reg file provided.

Configuring Certificate Information

A trusted Certificate Authority (CA) must provide the certificate for the Load Balancer server. The **Name** and **Issued To** fields for the Certificate must be the fully qualified domain name (FQDN) of the load balancer server. The certificate must be installed in the Internet Information Services (IIS) Manager on at least one of the Pipeline Pilot Servers.

Note: For more information about certificate requirements, see the Pipeline Pilot Administration Portal. Navigate to the **Security > SSL Certificate** page and choose the **Obtain a certificate from a recognized signing authority** option. Review the on-screen help for *Authority Signed Certificate* at the bottom of the page.

1. Create a .pfx (Personal Information Exchange File) as follows:
 - a. In the IIS Manager, double-click **Server Certificates**.
 - b. Right-click on the Certificate and select **View...**
 - c. On the **Details** tab, select the **Thumbprint** field and click **Copy to File...**
This opens the Certificate Export Wizard.
 - d. Click **Next** to start the export.
 - e. On the **Export Private Key** page, select **Yes, export the private key** and click **Next**.
 - f. Check the **Include all certificates in the certification path if possible** and **Export all extended properties** checkboxes and click **Next**.
 - g. Set a password for the file and click **Next**.
 - h. Click **Browse...** and select a folder where the .pfx should be saved. For example:
D:\Certificates\
 - i. Enter a **File name**, for example MyServer.pfx, and click **Save**. Click **Next**.
 - j. Click **Finish** to export the .pfx file.
2. In the Pipeline Pilot <pps_install>\install directory, create the following **ses** files:
 - ses-signed.crt
 - ses-signed.key
 - ses-signed.chain
3. Open a command prompt window and navigate to the <pps_install>\install directory.
4. Execute the following commands, ensuring the hyphens are correct (do not copy directly from this document):

```
openssl pkcs12 -in D:\Certificates\MyServer.pfx -out
D:\Certificates\ses-signed.crt -nokeys -nodes

openssl pkcs12 -in D:\Certificates\MyServer.pfx -out
D:\Certificates\ses-signed.key -nocerts -nodes

openssl pkcs12 -in D:\Certificates\MyServer.pfx -out
D:\Certificates\ses-signed.chain -chain -nodes
```

Replace the example D:\Certificates\MyServer.pfx and D:\Certificates\ses-signed.crt file paths with the correct locations for your server.

5. Copy the three **ses** files to the Pipeline Pilot install folder: <pps_install>\web\conf.

6. Also copy the three **ses** files to a network shared folder accessible to all other Pipeline Pilot Servers used in the load balancing configuration, for example `\\networkshares\loadbalancingconfig`.
7. Launch the Pipeline Pilot Administration Portal:
 - a. Login and navigate to **Security > SSL Certificate**.
 - b. Choose **Obtain a certificate from a recognized signing authority**.
 - c. Click **Save Option**.
8. Log out of the Pipeline Pilot Administration Portal.
9. Restart **Pipeline Pilot 2021 (Manager)** service.

Pipeline Pilot Server Administration Configuration

Note: If you are planning to register with Foundation Hub, it is recommended that you configure Authentication for the primary server before exporting the configuration to the load-balanced nodes. See the *Pipeline Pilot Administration Guide* for details about setting up Authentication.

1. Open the Pipeline Pilot Administration Portal and log in:
`https://<server1_name>.mycompany.com:9943/admin/`
2. Open the **Setup > Folder Locations** page and configure the following directories to the network folders you created:

User Directory	\\networkshares\shareduser
Jobs Directory	\\networkshares\sharedjob
Shared Public Directory	\\networkshares\sharedpublic
Local Temp Directory	\\networkshares\sharedjob
Upload Directory	
XMLDB Directory	

3. Click **Validate Locations**. This ensures the server can communicate with these directories.
4. Click **Save**.
5. Open the **Setup > Reverse Proxy and Load Balancing** page.
6. In the **Reverse Proxy Name and Aliases** section add the connection details of the load balancer server:
 - **Full Name:** `<lb_server>.mycompany.com`
 - **Aliases:** `192.168.2.100`
7. In the **Reverse Proxy Ports** section enter the HTTP and SSL port numbers used by Pipeline Pilot Server on the load balancer server. For example:
 - **HTTP:** 9944
 - **SSL:** 9943
8. Check the **Load Balanced** checkbox.
9. Click **Save**.

10. Open the **Maintenance > Export Import Configuration** page and click **Export Configuration**.
 - a. Select **Save File** and click **OK**.

Note: By default, the XML file is called `ServerConfig-<servername>.xml`.

- b. Open the **Downloads** folder and copy the server configuration file.
 - c. Navigate to a location on a network shared folder that can be accessed by other Pipeline Pilot Server in the load balanced configuration and paste the server configuration file, for example in `\\networkshares\loadbalancingconfig`.
 11. Open the **Setup > Data Sources** page and click **Export All Data Sources**.
 - a. Select **Save File** and click **OK**.
- Note:** By default, the exported file is called `DataSource.xmlx`.
- b. Open the **Downloads** folder and copy the server configuration file.
 - c. Navigate to a location on a network shared folder that can be accessed by other Pipeline Pilot Server in the load balanced configuration and paste the data sources export file, for example in `\\networkshares\loadbalancingconfig`.
 12. Log out of the Pipeline Pilot Administration Portal.
 13. Restart the **BIOVIA Pipeline Pilot 21.1.0 (Manager)** service. This will automatically restart the other two BIOVIA Pipeline Pilot services for Httpd and Tomcat.

Configuring Subsequent Pipeline Pilot Servers for Load Balancing

Repeat the following for each subsequent Pipeline Pilot Server added to the load balanced configuration.

IMPORTANT! Stop all Pipeline Pilot services on the existing load balanced Pipeline Pilot Servers so that you can configure load balancing on the other Pipeline Pilot Servers.

Setting up the Registry Keys (Windows)

1. In Windows, open the **Registry Editor**.
 - a. Navigate to **HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > services > LanmanWorkstation > Parameters**.
 - b. Right-click **Parameters** and select **Export**.
 - c. Export this registry entry as a back up for subsequent changes.
2. In Windows Explorer, navigate to the `<pps_install>\install\templates\win_noarch` folder.
3. Double-click on `LoadBalancedConfig_Enable.reg`. This will add the following information to the Windows Registry:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Lanmanworkstation\Parameters]
"FileInfoCacheLifetime"=dword:00000000
"FileNotFoundCacheLifetime"=dword:00000000
"DirectoryCacheLifetime"=dword:00000000
```

Notes:

If you are installing on load balanced Windows 2012 R2 Servers and have updated VMWare Tools and applied Microsoft Updates, you may encounter an issue where the Pipeline Pilot Server Administration Portal cannot be opened and protocols cannot be run. This issue is likely caused by the removal of needed registry entries during the update. To fix this re-run `LoadBalancedConfig_Enable.reg` as described above in step 3 to re-add them.

Pipeline Pilot Server Administration Configuration

1. Ensure that all three Pipeline Pilot services are running.
2. Open Windows Explorer and navigate to the network shared folder where the **ses** files for the primary Pipeline Pilot Server are stored, for example `\\networkshares\loadbalancingconfig`.
3. Copy these **ses** files to the Pipeline Pilot `<pps_install>\web\conf` directory.
4. Log into the Pipeline Pilot Administration Portal.
5. Open the **Maintenance > Export Import Configuration** page and click **Import Configuration**.
6. Navigate to the configuration file exported from the primary Pipeline Pilot Server and click **Upload**.
7. Open the **Setup > Data Sources** page and click **Import Data Sources**.
8. Navigate to the data source export file from the primary Pipeline Pilot Server, check the **Overwrite Duplicates** checkbox, and click **Upload**.
9. Log out of the Pipeline Pilot Administration Portal.
10. Restart the **BIOVIA Pipeline Pilot 21.1.0 (Manager)** service to restart all the Pipeline Pilot services.

Testing the Load Balanced Configuration

For each Pipeline Pilot Server in the load balanced configuration attempting to access the server when it is not available should automatically redirect to another load balanced server.

To test this you can simulate a server being unavailable by stopping its Pipeline Pilot Services, then an attempt to connect will access a different server that is available.

1. On the **server1** machine corresponding to `<server1>.mycompany.com`, ensure that all Pipeline Pilot services are stopped.
2. Open the Pipeline Pilot Client and set the **Active Server** to the load balancer server, (for example: `<lb_server>.mycompany.com`).
3. In the **Protocols** explorer, expand the **Web Services > Web Port Examples > Generic** folder and open the **Show Environment** protocol.
4. Run the protocol. This opens a report showing the *Server Name* for **server2**.
This confirms that, although **server1** is not running, the job has been sent to the load balancer and passed on to **server2** where it has been run successfully.
5. Close the report and close Pipeline Pilot Client on **server1**.
6. Start the **BIOVIA Pipeline Pilot 21.1.0 (Manager)** services on **server1**.

Repeat these steps for another server, for example:

1. On the **server2** machine corresponding to `<server2>.mycompany.com`, ensure that all Pipeline Pilot services are stopped.

2. Open the Pipeline Pilot Client and set the **Active Server** to the load balancer server, (for example: <lb_server>.mycompany.com).
3. In the **Protocols** explorer, expand the **Web Services > Web Port Examples > Generic** folder and open the **Show Environment** protocol.
4. Run the protocol. This opens a report showing the *Server Name* for **server1**.
This confirms that, although **server2** is not running, the job has been sent to the load balancer and passed on to **server1** where it has been run successfully.
5. Close the report and close the Pipeline Pilot Client on **server2**.
6. Start the **BIOVIA Pipeline Pilot 21.1.0 (Manager)** services on **server2**.

Repeat these steps for each Pipeline Pilot Server in the load balanced configuration. When there are two or more available Pipeline Pilot Servers in the load balanced configuration running the **Show Environment** protocol (or any other job) through the load balancer server will result the jobs being run of different individual Pipeline Pilot Servers depending on their resources.

For example, if you are testing **server1** (with its Pipeline Pilot services stopped) and the load balancer is configured with **server2** and **server3** then the job may run on either of these servers and running the job multiple times may result in the Server Name in the report being different between job runs. This indicates that the load balancer is performing correctly.

Chapter 9:

Installing on Grids and Clusters

This guide is for system administrators who want to configure Linux servers to deploy clustering or distributed grid computing with Pipeline Pilot.

Pipeline Pilot Clustering Overview

Pipeline Pilot clustering allows you to scale your Pipeline Pilot server to take advantage of multiple servers to distribute Pipeline Pilot protocols. Each node in the cluster includes a Pipeline Pilot server and shares the installation with the head node. The primary server acts as a dispatcher of Pipeline Pilot jobs to the other servers in the cluster. Multiple server machines access a single Pipeline Pilot installation on a shared file system and using the same protocol database (XMLDB).

Clustering Modes

Pipeline Pilot supports the following clustering modes:

- **Private Cluster:** The client connects to the head node and sends all jobs to the child nodes by the head node on a per-job basis. Private clustering works best for distributing jobs. Choose private clustering if you expect to run a lot of distributed or parallel subprotocols. This option dispatches jobs on a job-by-job basis without assigning a user to a particular node.
- **Public Cluster:** The client logs into the head node and then is assigned a child node for running jobs for the duration of the client session. Public clustering works best for distributing users. Choose public clustering if you have a number of users and would like to take advantage of a pool of available servers for nonparallel tasks.

Note: In both modes, jobs do not normally run on the head node or primary server.

Distributed Grid Computing Overview

Grid computing is a type of distributed computing in which CPU resources are shared across a computer network, allowing all connected machines to operate together. Pipeline Pilot supports technologies such as PBS Professional and Slurm. In addition, you can configure Pipeline Pilot to support the integration of other grid engines.

Distributed grid computing makes it possible to:

- Access extra computing power to work more efficiently.
- Queue long-running jobs.
- Efficiently distribute protocols over cluster nodes.
- Parallelize a protocol and distribute subsets of the data efficiently over cluster nodes.
- Take advantage of improved capabilities for computer hardware administration and performance tuning.

With grid technology, a large number of protocol runs can be distributed to a compute cluster, providing efficient use of enterprise compute resources. You can also queue specific long-running protocols, optimizing the turnaround for critical short-running protocols. Overall, this allows a community of Pipeline Pilot users to run more protocols with maximum efficiency for each individual job.

Parallel Computing

With parallel computing, you can efficiently distribute individual jobs to cluster nodes, making it possible to tackle extremely large data sets not feasible on a single server. You can run any subprotocol in parallel on a specified set of remote servers and specify batch size for data records, remote hosts, and number of simultaneous jobs. Optionally, you can run these parallel subprotocols under a distributed grid computing engine, giving you more options for job management and load balancing. See the *User and Host* example protocol.

Pipeline Pilot Servers with Grid-enabled Clusters

A Pipeline Pilot server can function as an edge node for a grid-enabled cluster. In this deployment setup, you can run protocols on the server or distribute them to a grid-enabled cluster that supports parallelization at the process or record level (coarse-grained). This deployment method is optimal when data can be split into smaller chunks and processed independently.

Notes:

- All protocols that are not configured to run on the grid will run on the server.
- If your throughput decreases when handling many quick-running jobs on the server, you can scale by using a Pipeline Pilot cluster (described below).

Requirements

For system requirements for Pipeline Pilot and supported grid engines, see the *Pipeline Pilot System Requirements* for a list of supported grid engines.

Requirements for Clustering

A Pipeline Pilot cluster consists of a set of compute nodes that share a mounted file system on which you install the Pipeline Pilot software. You designate one node as the primary node—all others are secondary nodes. When configured this way, all clients for the cluster connect to the primary node, while end users do not require any knowledge of the other nodes.

For Pipeline Pilot clustering:

- All nodes must have access to the shared file system where you install the Pipeline Pilot software.
- The path to the Pipeline Pilot installation must be identical for all nodes. (For example, the path `/opt/biovia/plpserver` is valid for all nodes, primary and secondary.)
- All client machines running the Pipeline Pilot client software should have HTTP access to the primary node. If client machines do not have access to the secondary nodes, you can set the **Private Cluster** option in the Pipeline Pilot Admin Portal. For more information, see [Requirements](#) on page 52.
- All nodes should have HTTP access to the primary node for requesting web services.
- The primary node should have HTTP access to all secondary nodes for requesting web services.
- A local (non-NFS) directory is required on each node with write access for lock file management. The default path is `/var/tmp`—the installer allows you to specify a location that must exist on all nodes with Read, Write, and Execute permissions for all users.
- Each server node should be installed with the operating system packages as specified in the Pipeline Pilot system requirements for Linux servers. For more information, see the *Pipeline Pilot Server Install Guide* and the Pipeline Pilot System Requirements.

Best Practices

- Ensure that you have a set of designated servers that are networked to allow HTTP access to each other. Select one node as the primary node—it serves as the point-of-contact from Pipeline Pilot clients. The primary node requires HTTPS access from client machines.
- Ensure that all servers have identical access to a shared file system and that it is accessible from the same path on all servers.
- For large clusters, the primary node should not be included in the list of job-running nodes, because it cannot delegate a job to itself. This node supports various services for the cluster as a whole, such as secure login, job delegation, file system browsing, shared component and protocol database (XMLDB) service. The execution of compute-intensive jobs on the primary node can compromise the performance of the cluster as a whole, and outweigh the advantages of including an extra computational node for job execution. With a small cluster, the balance favors making this node play its part both as a job runner and as a primary node.
- The system hardware and software requirements for each node in the cluster are the same as for any server. For details, see the *Pipeline Pilot Server Install Guide*.

Requirements for Grid Engines

A grid installation is virtually identical to a cluster installation. The difference is that instead of running Pipeline Pilot on the cluster nodes, the grid engine software manages the cluster nodes. See the *Pipeline Pilot System Requirements* for a list of supported grid engines.

Notes:

- You can configure Pipeline Pilot to work with a custom grid engine. For details, see [Requirements](#) on page 52.
- Pipeline Pilot must be installed on a Linux machine that can support your distributed grid computing system. There are no additional hardware requirements beyond what is specified for a normal installation.

Pre-Installation for a Grid or Cluster

Perform these steps before installing Pipeline Pilot on a grid or cluster.

User Accounts

1. Create user and group account to install and run the server.
 - The group must only contain the user. Do not add other users.
 - This can be a no-login account. The account must be accessible from all machines on the cluster.
2. Create an installation directory.
 - Create `/opt/biovia/plpserver` on a local disk, not a share. The `/opt/biovia` folder is for installation of BIOVIA products on the cluster. The Pipeline Pilot server is installed in `/opt/biovia/plpserver`.
 - Change this directory's owner to the user and group you created:

```
mkdir -p /opt/biovia; chown ppuser.ppgroup
/opt/biovia
```

NFS Configuration

For a clustered or grid environment, the primary server needs to export the installation to the other nodes in the cluster. This step is critical for the proper operation of the server.

IMPORTANT! Use these settings for mounting the Pipeline Pilot server directory. They minimize caching while providing good performance.

1. Increase the number of available NFS daemons on the server.
 - Edit `/etc/sysconfig/nfs` or `/etc/init.d/nfs` and change the value of `RPCNFSDCOUNT` to 32 or higher if you plan to run many jobs on the grid from Pipeline Pilot.
 - Restart NFS or reboot:

```
systemctl restart nfs
```

or

```
/etc/init.d/nfs restart
```

2. Export the installation directory. Edit `/etc/exports` and add entries on the target machines for exporting the installation. Use the `sync` and `no_root_squash` options. For example:

```
/opt/biovia \
node1(rw,no_root_squash,sync) \
node2(rw,no_root_squash,sync) \
node3(rw,no_root_squash,sync) \
```

After adding all of the nodes, export the file systems:

```
exportfs -a
```

3. Log into each node that you exported to and mount the file system you exported.

```
ssh node1
```

```
mkdir /opt/biovia
```

```
vi /etc/fstab
```

```
ppserver:/opt/biovia /opt/biovia nfs
rw,hard,intr,lock,nfsvers=3,rsiz=2048,wsiz=2048,acdirmin=1,acdirmax=1,
acregmin=0,acregmax=0,noatime
```

4. Mount the file system and make sure it works.

```
mount /opt/biovia
```

Reducing NFS Dependencies for Grids and Clusters

You can reduce the amount of NFS traffic generated by running Pipeline Pilot on a grid or cluster. To improve performance for disk-intensive protocols, you can use the **Local Temp Directory** setting is available in **Pipeline Pilot Admin Portal > Setup > Folder Locations**.

The default temporary file folder uses the same location as the jobs directory. The global variables `LocalJobTempDirectory` and `TempDir` indicate the same folder location. You can specify a local disk resource for temporary files that are not needed for the final result of a calculation. The path specified must exist and be writeable on all nodes of the cluster. Files placed in this directory are removed after a protocol run.

The default setting uses the same location as the job temporary directory `@TempDir`. Specifying `/tmp` works for this setting as well as any local scratch space you configured. Sort and Merge components use this setting if specified.

The global variable associated with this setting is named `@LocalJobTempDirectory`, which can be accessed and used in protocols. Using this global in conjunction with the Local Temp Directory setting in the Pipeline Pilot Admin Portal improves performance and reduces NFS traffic when running on a busy cluster or grid.

Note: When running some applications in this way, you may need to reset this folder to a local path that is valid for each node of the cluster. Each job running on the cluster creates a job-specific subfolder for temporary files, accessible in the protocol by the global variable `LocalJobTempDirectory`.

To relocate the Local Temp Directory:

1. From the Pipeline Pilot Admin Portal, open **Setup > Folder Locations**.
2. Set **Local Temp Directory** to the new folder location.
3. Click **Validate Locations**.
4. Click **Save**.

Tip: For best results, your server should be idle when you make this change.

Security

When you install the server, run `scirootinstall`, even if you do not plan on using impersonation. Perform the following tasks before installing Pipeline Pilot:

1. If you plan on using local accounts, change the permissions on `/etc/shadow` so it can be read.

```
chgrp ppgroup /etc/shadow
```

```
chmod g+r /etc/shadow
```

2. For a machine with SELinux enabled, run `sciseallow`, which installs a module to allow Pipeline Pilot to run. You do not have to disable SELinux to run Pipeline Pilot.

IMPORTANT! Run this script before installing Pipeline Pilot on the server.

Grid Engine Setup

When using a grid engine such as PBS or Slurm, make sure you are able to submit jobs from the host where you are installing Pipeline Pilot. The Pipeline Pilot server does not have to be installed on the head node of the grid, but it must be installed on a node that can submit jobs to the grid. It is recommended that you use edge node configuration. In this type of configuration the Pipeline Pilot server can submit jobs to the grid, but is not a member of the grid itself.

Pre-Installation Checklist

It is important that you thoroughly review the following checklist to ensure that you perform all preliminary tasks before deploying a Pipeline Pilot cluster or grid. This checklist applies to both configurations.

1. If you are installing on an SE (Secure Extensions) Linux enabled machine, run `sciseallow` as root before installing Pipeline Pilot.
2. Configure a user and a group account for running Pipeline Pilot. This account should also be available on the cluster. The Pipeline Pilot user account does not require a login (SU-only account). Normal users should not be members of this group.
3. To increase performance and minimize I/O bottlenecks, increase the number of NFS daemons

running on the primary Pipeline Pilot server.

Log in as root and edit the file `/etc/sysconfig/nfs` or `/etc/init.d/nfs`.

Change:

```
RPCNFSDCOUNT=8
```

To:

```
RPCNFSDCOUNT=32
```

Restart the NFS service:

```
# /etc/init.d/nfs restart or # systemctl restart nfs
```

4. Ensure that you have sufficient local disk space to install the Pipeline Pilot software and data.

Install the primary Pipeline Pilot server on a disk that is local to the machine that runs it. Installing on an NFS-mounted disk causes protocol failures and other issues.

IMPORTANT! The path for the mount point must match on all nodes in the cluster. If you install Pipeline Pilot in `/opt/biovia`, then all cluster nodes need to access the Pipeline Pilot installation as `/opt/biovia`.

5. Create the mount point for the file system.

```
mkdir /opt/biovia
```

```
# chown ppuser:ppgroup /opt/biovia
```

6. Export the mount point for the file system. Edit the file `/etc/exports` and add entries for each node you want to export to. Also, set the `no_root_squash` option.

```
/opt/biovia \
Node1(rw,no_root_squash,sync) \
Node2(rw,no_root_squash,sync)
# exportfs -a
```

7. Mount the exported file system for each node in the cluster.

Edit the `/etc/fstab` file and add an entry for the exported file system using the following options:

```
ppserver:/opt/biovia /opt/biovia nfs
rw,hard,intr,lock,nfsvers=3,rsize=2048,wsz=2048,acdirmin=1,acdirmax=1,
acregmin=0,acregmax=0,noatime
# mount /opt/biovia
```

8. If you are installing on a grid, verify that you can submit a simple job to the grid engine from the Linux server Pipeline Pilot is installed on and that it runs successfully.

Manually run a job through the system and verify that it works. This usually involves using the grid engine submit command.

- PBS: `qsub`
- Slurm: `sbatch`

9. If you are using R Statistics or Discovery Studio, plan on installing these packages in a location that can be exported to the cluster with a single share.

For example, to install both R and Discovery Studio and export them with one mount, install Pipeline Pilot in `/opt/biovia/plpserver`, install R in `/opt/biovia/rstats`, and install Discovery Studio in `/opt/biovia/ds`. This allows you to export `/opt/biovia` so the cluster can access and run all the applications.

For details see the *R Software for Pipeline Pilot Installation and Configuration Guide*.

Installation Guidelines

1. Log into the user account that you configured and install Pipeline Pilot server.
2. Run the `scirootinstall` script as part of the installation. This allows for impersonation and automatically configures the grid node setup script to use the Pipeline Pilot user account. Run `scirootinstall` even if you do not plan on using impersonation.
3. After installation, check that you can connect to the server portal. In a web browser, navigate to `http://<servername>:<port>`, where `<servername>` is the name of your Pipeline Pilot server. The default port is 9944.
4. Test the Pipeline Pilot server installation to ensure that everything works as expected. You can run server diagnostic tests from the Pipeline Pilot Home Page. For more information, see the *Pipeline Pilot Admin Portal Guide*.
5. Perform post-install tasks. For details, see [Post-Installation for a Grid or Cluster](#) on page 57.

Tip: You can also download a client from the server portal and run a protocol from a remote computer back to the newly-installed server.

Post-Installation for a Grid or Cluster

After the installation is complete, run `scirootinstall` and configure your server as follows:

1. Shut down the server. The installer automatically starts the server, but you still need to perform a few tasks before you can use it.


```
cd /opt/biovia/plpserver/linux_bin
./stopserver
```
2. Run `scirootinstall` to assign permissions, add init scripts, set up pam, and configure the server to run with impersonation. Run this script even if you do not intend to use impersonation.
3. If using impersonation:
 - a. Run `scigrgridsetup` on each exec node. `scigrgridsetup` creates the mutex directory and lock file, which can have problems with permissions if created by an impersonated user.
 - b. Remove the output files for the grid engine in the Pipeline Pilot temporary directory:


```
rm -f temp/*qsub.out
```
 - c. Define the group bit on directories. This applies when using alternative locations for the users, jobs, or the XMLDB. This allows your server to access files created by impersonated users. `scirootinstall` does this for the default installation. Perform this step if you use an alternate location.


```
mkdir Altuser
chown ppuser.ppgroup Altuser
chmod 2775 Altuser
```
4. For a grid, run `scigrgridsetup` on each node that you plan to use to run Pipeline Pilot protocols. This script adds an init script for Xvfb and runs `sciseallow`.

IMPORTANT! If you do not run Xvfb, components that use R software to generate plots and graphics may not work. For more information, see *R Software for Pipeline Pilot Installation and Configuration Guide*.

Configuring your Linux Server

1. Restart the server.

```
cd /opt/biovia/plpserver/linux_bin
```

```
source ppvars.sh
```

```
./startserver
```

2. Open the Pipeline Pilot Admin Portal and configure the options you want to use. Enable impersonation and verify that it works before making other configuration changes.
3. For clustering support, log into each of the cluster nodes and start the servers. Add the nodes to the cluster node list and verify that the nodes are active (green). Round robin is recommended.

```
ssh node1
```

```
cd /opt/biovia/plpserver/linux_bin
```

```
source ppvars.sh
```

```
./startserver
```

Configuring a Cluster

After installing the primary server, set up the nodes on the cluster.

1. Log into the node and change the directory to the Pipeline Pilot installation on the shared file system. From `linux_bin`:

```
./startserver
```

2. Open the Pipeline Pilot Admin Portal on the primary server.
3. Choose **Setup > Clustering**.
4. From the **Load Balancing** list, select an option:
 - **Round Robin:** Jobs are assigned to each server in sequence. This option is recommended.
 - **Job Leveling:** Servers with the fewest jobs are prioritized.
5. Specify the **Internal Server Name** and click **Save**. This is the name of the head node of your cluster as seen from inside the cluster.
6. To define your cluster as private (endpoints are only accessible from the primary server and not from client machines), check **Private Cluster**.

Notes:

Private clustering is required under the following scenarios:

- If the primary server is the only server that is accessible directly from the client machines. It acts as a conduit for all communication to other servers in the cluster.
- When using the cluster behind a reverse proxy. When public clustering is used, the client must be able to directly access the runner endpoints.

7. Add runner endpoints. Specify the names of all hosts for Pipeline Pilot Servers that are part of the cluster. Ensure these servers are running and active. Enter the server name in **Cluster Runner Endpoints** in the form <servername> : <port>, and then click **Add**.
8. (Optional) You can configure your cluster to store temporary job files to a file system that is local to the running job. This is useful when the directory is located on a shared resource, such as an NFS mount. To configure, go to **Setup > Folder Locations** and set the **Local Temp Directory**.

Tip: Local temporary storage can improve protocol performance and reduce network loads on clusters and grids.

9. Rerun your initial installation checks on the primary node to ensure that the secondary nodes run jobs.
 - a. From the Pipeline Pilot Server Home Page, click **Server Diagnostic Tests**.
 - b. Run simple jobs and check that the server name for the executing job is different each time you run.

IMPORTANT! If you plan to run with impersonation and authentication enabled in the cluster, copy the primary node's `/etc/pam.d/scitegic` file to the nodes.

Testing the Cluster

Rerun your initial installation checks on the primary node to ensure that the secondary nodes run jobs.

1. From the Pipeline Pilot Server Home Page, click **Server Diagnostic Tests**.
2. Run simple jobs and check that the server name for the executing job is different each time you run.
3. Review the **Server** column in **Reports > Completed Jobs** so see which node ran the job.

Configuring a Grid

For a grid, configure the nodes before the main server.

Configuring Nodes

For each grid cluster node that runs Pipeline Pilot protocols, mount the Pipeline Pilot installation and configure the node to run Pipeline Pilot.

1. Log into each node and become the root user.

```
ssh node1
```

```
$ sudo -s
```

2. Create a directory for the mount point.

```
# mkdir /opt/biovia
```

3. Add a line to `/etc/fstab` for the primary Pipeline Pilot server.

```
pphost:/opt/biovia /opt/biovia nfs rw, hard, intr, lock, nfsvers=3,
rsize=2048, wsize=2048, acdirmin=1, acdirmax=1, acregmin=0, acregmax=0,
noatime
```

4. Mount the Pipeline Pilot directory.

```
# mount /opt/biovia
```

5. If configuring an SE Linux-enabled server, run the `sci seallow` script. For more information, see [Security](#) on page 55.

Defining Grid Cluster Queues for Nodes

If you configured a subset of your grid cluster nodes to run Pipeline Pilot protocols, define one or more cluster queues for the nodes. This ensures that protocols only run on nodes that are enabled to process them. For more information, see the documentation for your grid engine product.

Configuring the Server

After you configure the participating grid cluster nodes to run Pipeline Pilot, enable the grid on the primary server.

1. Go to **Setup > Grid**.
2. Set Grid Engine Configuration options:
 - **Grid Engine Type:** Select the grid engine (SGE, PBS, LSF, and CUSTOM). For more information, see [Custom Grid Engines](#) on page 61.
 - **Grid Engine Path:** Set the absolute file system path to the grid engine directory.
 - **Internal Server Name:** Specify the host name or IP address for your Pipeline Pilot Server as seen by the nodes in the grid. If your grid has a high-speed network that connects the nodes, you can take advantage of this by specifying the IP address or name for this network interface.
 - **Grid Engine Default Queue:** Configure the default queue:
 - If Pipeline Pilot is installed on a subset of the available grid cluster nodes, enter the default queue name. If a client does not specify a queue name, this queue is used to process the job.
 - To use the grid engine default, leave this option blank. The client will not specify a queue name and the grid engine default queue is used (no -q argument to qsub).
 - **Grid Engine Submit Options:** Configure options to be submitted to the grid engine by default. These options provide generic grid engine fields that are translated into the correct command line parameters based on the selected grid engine. The following options can be defined:
 - **Account:** Equivalent to Project, Allocation, or Account depending on the grid engine.
 - **Maximum Execution Time:** Maximum time to allow the grid engine job to execute in HH:MM:SS, hours, minutes, seconds. LSF ignores seconds.
 - **Parallel Environment:** Used by SGE and UGE grid engines. It is required if **Number of Cores** is specified. Other Grid Engines ignore this value.
 - **Number of Cores:** Number cores the protocol will run on.
 - **Other:** Pass additional parameters directly to the grid engine as is. These options are not translated.

Note: These values will be overridden by grid options specified by protocol and subprotocol parameters.

- **Maximum Number of Grid Jobs that can be submitted by a protocol:** Maximum number of grid jobs that may be submitted on this server to support EACH individual parallel processing subprotocol. If the value is blank the equivalent value for normal Pipeline Pilot jobs will be used.
- **Maximum Number of Grid Job Submission failures:** Maximum number of grid job submission failures that can occur on each Batch Thread Manager prior to canceling the subprotocol submitting them. The default number is 5. Note that the number of Batch Thread Managers is highly dynamic and determined at run time, so the actual number of grid job submissions will probably be more than this setting, perhaps many more.

- **Run On Grid by Default:** Run all but blocking client jobs on the grid. Note that some quick running jobs always run on Pipeline Pilot Server regardless of this setting.

3. Click **Save**.

Custom Grid Engines

When configuring Pipeline Pilot to work with your grid, you can specify a custom grid engine in the Pipeline Pilot Admin Portal. The following scripts are available in `<pps_install>/linux_bin` to support grid engine functionality:

- `custom_submit`: Invokes `qsub`.
- `custom_status`: Invokes `qstat`.
- `custom_kill`: Invokes `qdel`.

These scripts are based on UGE engine scripts. To configure Pipeline Pilot to support your grid engine, modify these scripts so they work with your specific grid engine. For more information, see the documentation for your grid engine product.

Impersonation

If you need to track grid usage by user account, enable impersonation of user accounts. For more information, see *Impersonation* in the *Pipeline Pilot Admin Portal Guide*.

Testing a Grid Engine

1. From the Pipeline Pilot client, open the example protocol *Grid Example 1*.
2. At the protocol level, enable the **Run On Grid** parameter.
3. Open the **Run on Grid** parameter group and select the queue to run your job. If you do not specify a queue, the grid engine default is used.
4. Run the protocol. The status bar displays the following messages:
 - **Queued:** Your job is submitted to the grid and is waiting for processing.
 - **Connected:** Your job is running on the grid. This message is not always displayed when you submit a job to the grid, because each grid has its own queuing latency.
5. If you created multiple queue names, run the protocol on each queue.

Appendix A:

Unattended Installation

Note: The following example details an unattended Linux installation, but the procedure also applies to a Windows installation.

IMPORTANT! `<pps_install>` is the root of the Pipeline Pilot installation.

On Windows this is typically:

C:\Program Files\BIOVIA\PPS

On Linux this depends on the location of your Pipeline Pilot installation, but may be:

[Home]/BIOVIA/PPS

Setting Required Pre-Installation Environment Variables

The following parameters must be set before you begin an installation, upgrade, or reinstall, or before add/remove packages begins.

IMPORTANT! You must be running in the bash shell.

Tip: The **red** text should be replaced by the appropriate value for your installation environment.

```
ROOT="<pps_install>"
export ROOT
```

```
SCITEGIC="$ROOT"
export SCITEGIC
```

```
PACKAGEDIR="apps/scitegic/core/packages_linux64"
export PACKAGEDIR
```

```
ODBCINST="$ROOT/$PACKAGEDIR/datadirect/odbcinst.ini"
export ODBCINST
```

```
MEDIADIR="/tmp/mydir"
```

Tip: Navigate to the bin directory, enter pwd, and then copy that value to the MEDIADIR variable.

Note: All the directories and files in this path must have execute permissions.

```
export=MEDIADIR
```

```
LD_LIBRARY_PATH="$ROOT/linux_
bin
":"$MEDIADIR":"$MEDIADIR/bin":"$ROOT/$PACKAGEDIR/
datadirect/lib":"$ROOT/apps/scitegic/openssl/lib/linux64"
export LD_LIBRARY_PATH
```

```
PERL5LIB="$ROOT/$PACKAGEDIR/perl/perl-5.26.1"
export PERL5LIB
```

```
SCITEGICMUTEXES=/var/tmp/scitegic-mutexes-9943-myuser
export SCITEGICMUTEXES
```

Note: The mutexes directory must have 777 permissions assigned on Linux and read-write access on Windows. On Windows you could use C:\Program Files\BIOVIA\PPS\temp, for example.

Installation Command Syntax

When you run the installer, use a command with this syntax:

```
"$MEDIADIR/installcmdline" -execute -d "$ROOT" -l "<license_path>" -p <HTTP_
port> -s <HTTPS_port> -c "<cluster_method>" -m "$SCITEGICMUTEXES" -mode
<install_mode> -pi <product>
```

Note: For information on \$MEDIADIR, \$ROOT, and \$SCITEGICMUTEXES see the [pre-installation parameters](#).

Where:

- \$MEDIADIR is the directory containing the Pipeline Pilot installer, that you set previously.
- \$ROOT is the directory of the Pipeline Pilot installation that you set previously.
- <license_path> is the path to the Pipeline Pilot license file.
For example: -l "/tmp/license"
- <HTTP_port> is the HTTP port for Pipeline Pilot to use.
For example: -p 9944
- <HTTPS_port> is the HTTPS port for Pipeline Pilot to use.
For example: -s 9943
- <cluster_method> is the clustering method to use.
Options are:
 - round-robin
 - job_leveling
 - no -c option: no clustering
 For example: -c round-robin
- \$SCITEGICMUTEXES is the mutex directory that you set previously.
For example: -m "\$SCITEGICMUTEXES"
- <install_mode> is the type of installation to perform.
Options are:
 - Upgrade
 - FirstTimeInstall
 - Reinstall
 - AddRemove
 For example: -mode FirstTimeInstall
- <product> is the name of a package to install in the format <vendor>/<package>, all packages provided with Pipeline Pilot by BIOVIA use scitegic as the vendor.
For example: -pi scitegic/imaging

Package Names

Notes:

- The following packages include collections that may not be installed with the regular Pipeline Pilot license.
- This list does not include packages from applications.

- scitegic/admetox
- scitegic/admetoxdata
- scitegic/admin
- scitegic/adminportal
- scitegic/allotrope
- scitegic/analyticalinstrumentation
- scitegic/apidocs
- scitegic/appcatalog
- scitegic/binstore
- scitegic/bioutilities
- scitegic/cassandra
- scitegic/chemistry
- scitegic/chemistryadv
- scitegic/chemistrycart
- scitegic/chemistrycontent
- scitegic/chemistrydata
- scitegic/chemistryhub
- scitegic/chemistrysdk
- scitegic/chemistrytlkt
- scitegic/chemmining
- scitegic/clientsdk
- scitegic/clientsdkdotnet
- scitegic/clientsdkjava
- scitegic/clientsdkjavascript
- scitegic/cloudview
- scitegic/compose
- scitegic/core
- scitegic/coreutils
- scitegic/dataaccess
- scitegic/discoverant
- scitegic/docs
- scitegic/draw
- scitegic/dsbuilder

- scitegic/editsvc
- scitegic/gems
- scitegic/gemsapp
- scitegic/geneexpression
- scitegic/geneexpressiondata
- scitegic/generic
- scitegic/hazelcast
- scitegic/hub
- scitegic/imaging
- scitegic/imagingadv1
- scitegic/imagingdata
- scitegic/installer
- scitegic/integration
- scitegic/javaserver
- scitegic/jupyter
- scitegic/kafka
- scitegic/labelprinting
- scitegic/launcher
- scitegic/listmgmt
- scitegic/massspecprot
- scitegic/massspecprotdata
- scitegic/materials
- scitegic/modeling
- scitegic/modelingadv
- scitegic/mongodb
- scitegic/notebook
- scitegic/openssl
- scitegic/pipettesketcher
- scitegic/plateanalytics
- scitegic/platform
- scitegic/polymerproperties
- scitegic/ppclient
- scitegic/ppwebclient
- scitegic/projectdb
- scitegic/queryservice
- scitegic/report
- scitegic/report2
- scitegic/reportadv
- scitegic/rss

- scitegic/rstats
- scitegic/runtimesvc
- scitegic/scheduler
- scitegic/scmobile
- scitegic/scnotebook
- scitegic/seqanal
- scitegic/seqanaldata
- scitegic/sharepoint
- scitegic/sketcherintegration
- scitegic/study
- scitegic/systemlibs
- scitegic/tensorflow
- scitegic/textanalytics
- scitegic/threedspace
- scitegic/waf
- scitegic/web
- scitegic/webgl
- scitegic/websvc
- scitegic/widgetbuilder

Installation Command Example

Note: A space is required between each package name when installing multiple packages at the same time.

To run the installer for a new, first-time installation the following command is entered. This example performs a default Pipeline Pilot installation.

```
"$MEDIADIR/installcmdline" -execute -d "$ROOT" -l "/tmp/license" -p 9944 -s 9943 -c round-robin -m "$SCITEGICMUTEXES" -mode FirstTimeInstall -pi scitegic/admin -pi scitegic/adminportal -pi scitegic/appcatlog -pi scitegic/clientsdk -pi scitegic/clientsdkdotnet -pi scitegic/clientsdkjava -pi scitegic/clientsdkjavascript -pi scitegic/core -pi scitegic/coreutils -pi scitegic/dataaccess -pi scitegic/docs -pi scitegic/dsbuilder -pi scitegic/gems -pi scitegic/gemsapp -pi scitegic/generic -pi scitegic/integration -pi scitegic/javaserver -pi scitegic/launcher -pi scitegic/mongodb -pi scitegic/queryservice -pi scitegic/report -pi scitegic/reportadv -pi scitegic/rss -pi scitegic/scheduler -pi scitegic/sharepoint -pi scitegic/waf -pi scitegic/web -pi scitegic/websvc
```


Appendix B:

Required RPMs

The following RPMs are required for Linux installations. The Pipeline Pilot Server distribution includes `check_system.sh`, which you can use to check which libraries may be required for your system. Note that the libraries must be the 64-bit version.

Red Hat 8.1+ and CentOS 8.1+

- `libnsl`

Red Hat 7.5+ and CentOS 7.5+

- `alsa-lib (x86_64)`
- `audit-libs (x86_64)`
- `compat-libstdc++-33 (x86_64)`
- `compat-libtiff3 (x86_64)`
- `e2fsprogs-libs (x86_64)`
- `expat (x86_64)`
- `glibc (x86_64)`
- `keyutils-libs (x86_64)`
- `krb5-libs (x86_64)`
- `libgcc (x86_64)`
- `gd (x86_64)`
- `libICE (x86_64)`
- `libpcap (x86_64)`
- `libselenium (x86_64)`
- `libsepol (x86_64)`
- `libSM (x86_64)`
- `libstdc++ (x86_64)`
- `libX11 (x86_64)`
- `libXau (x86_64)`
- `libXdmcp (x86_64)`
- `libXext (x86_64)`
- `libXi (x86_64)`
- `libXmu (x86_64)`
- `libXpm (x86_64)`
- `libXrender (x86_64)`
- `libXt (x86_64)`
- `libXtst (x86_64)`

- pam (x86_64)
- cairo (x86_64)
- fontconfig (x86_64)
- glib2 (x86_64)
- bzip2-libs (x86_64)
- freetype (x86_64)
- libjpeg-turbo (x86_64)
- zlib (x86_64)
- libdrm (x86_64)
- libXdamage (x86_64)
- libXfixes (x86_64)
- libXxf86vm (x86_64)
- libXinerama (x86_64)
- tcsh
- bc
- glibc

Appendix C:

Installing GPL-licensed Biology Tools

The Sequence Analysis collection makes use of third-party tools commonly used in the bioinformatics community. These tools provide common analyses and their implementation illustrate integration. Although they are part of the Sequence Analysis collection, many of these tools are GPL-licensed and must be installed separately. For details about the tools included with this installation, see Third-Party Tools in the *Sequence Analysis Guide for Pipeline Pilot*.

Windows

To install GPL-licensed Biology tools on Windows:

1. Download and extract `BIOVIA_2021.BIOVIABiologyGPL2021_win64.zip` to a folder such as `C:\BIOVIA\DS`.
2. In the extracted `BIOVIABiologyGPL2021_win64\bin` directory, double-click `scitegicsetup.exe` to run the installer.
3. The installer will detect that you already have a Pipeline Pilot installation. Click **Add/Remove Products**.
4. The license you used when you installed the server is verified. Continue using this license—click **Next** without entering the location of the license file again.
5. Ensure that the **Pipeline Pilot Sequence Analysis Collection** package is selected (this is required by Pipeline Pilot Server).
If you have additional Biology Collection packages available in your license and they are already installed on your Pipeline Pilot Server, ensure the associated third-party GPL tools packages are selected.
6. **Running Applications > Stop Services:** Click the **Stop Services** to stop the Pipeline Pilot Windows services. They will be restarted after installation.
7. **Assign Ports:** Use the same ports you specified for the server.
8. **Confirm Apache Login:** Accept the current configuration.

Linux

To install GPL-licensed Biology tools on Linux:

1. Download and extract `BIOVIA_2021.BIOVIABiologyGPL2021_Linux64.tgz`:

```
tar -xpvzf BIOVIA_2021.BIOVIABiologyGPL2021_Linux64.tgz
```
2. In the top level extracted folder, run `sciinstall` as a non-root user to run the installer.
3. The installer will automatically run the **Add/Remove packages** option and prompt you to enter the target directory for your Pipeline Pilot Server. Enter the same directory you specified for the server during the server installation steps.
4. Press **Return** when prompted for the license.
5. When prompted to **Choose a Collection number to toggle its action from display**, press **Return** to accept the **[y]** default:

- Ensure that the action for **Pipeline Pilot Sequence Analysis Collection** package is set to **Install**.
 - If you have additional Biology Collection packages available in your license and they are already installed on your Pipeline Pilot Server, ensure the associated third-party GPL tools packages are selected.
6. Continue with the installation. When prompted to **Start the server**, choose **y**.

Appendix D:

Installing BIOVIA License Pack for Materials Studio Collection

The Materials Studio Collection (MSC) components are licensed using the BIOVIA License Pack and the MSC requires the latest version of the License Pack. This can be obtained alongside the Materials Studio installation artifacts.

Note: If you have already installed Materials Studio 2021 on the machine you are using as your Pipeline Pilot server you do not need to install the License Pack again.

1. Obtain the correct version of Materials Studio for your platform.
2. Run the EXE or TGZ file to decompress the installable files.
3. Locate the **LicensePack** directory containing the installers. Run the installer in the appropriate manner for your system:
 - **Linux:** `./lp_setup`
 - **Windows:** `lp_setup.exe`

IMPORTANT! After installation of the License Pack it should be configured according to the instructions in the "Licensing Materials Studio" chapter of the *BIOVIA Materials Studio 2021 Installation and Administration Guide*.

Linux Post-Installation Instructions

On Linux, once you have installed BIOVIA License Pack for use by Materials Studio Collection in Pipeline Pilot, you need to configure its location.

1. Open the Pipeline Pilot Admin Portal and select the **Setup > Global Properties** page.
2. Set the Package to **BIOVIA/Materials Studio**.
3. Set the **LicensePackDir** property to the directory where the BIOVIA License Pack is installed.
4. Click **Update**.

Dassault Systèmes Support Resources

For additional resources or to contact Dassault Systèmes Customer Support, visit the Support portal:

<https://www.3ds.com/support/>

From this portal, you can:

- Call or email Dassault Systèmes Customer Support
- Submit a request
- Download installers
- Access hardware and software requirements
- Access Knowledge Base
- Access Communities and Twitter feeds