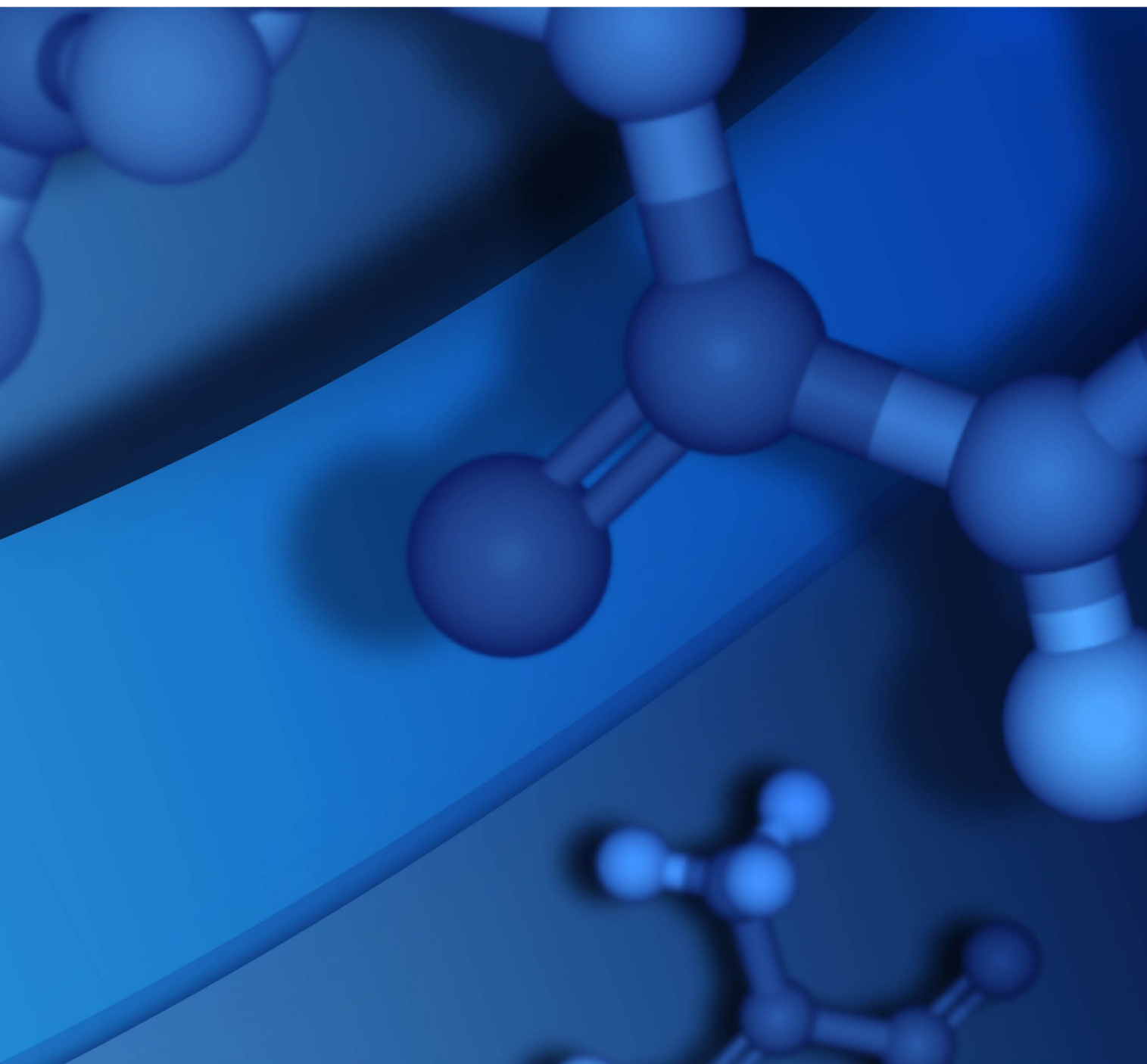


ADMIN PORTAL GUIDE

PIPELINE PILOT 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 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: Introduction | 1 |
| Pipeline Pilot Server Overview | 1 |
| Supported Operating Systems | 1 |
| Client Software | 1 |
| Developer Tools | 1 |
| Enterprise Architecture | 1 |
| Protocol Database Sharing and Data Access | 1 |
| Security | 2 |
| Java Server Package | 2 |
| About this Guide | 2 |
| Getting Additional Information | 2 |
| Additional Information | 3 |
| Getting Started with BIOVIA Pipeline Pilot Server | 3 |
| Administration Portal Overview | 3 |
| Accessing the Admin Portal | 3 |
| Admin Portal Home Page Features | 4 |
| Admin Pages Explorer | 4 |
| Displaying the Admin Pages Explorer | 5 |
| Client Support | 6 |
| Overview of Client Types | 6 |
| Client Installation Overview | 7 |
| Automatic Client Software Updates | 7 |
| Confirming User Identity | 8 |
| BIOVIA Desktop Connector | 8 |
| Server Home Page | 9 |
| Accessing the Home Page | 10 |
| Managing Administrator Access | 11 |
| Web Services Overview | 11 |
| Client SDK Access | 12 |
| SOAP Web Services API | 12 |
| RESTful Web Services API | 12 |
| Apache 2 HTTP Server | 12 |
| Windows Apache Users | 13 |

| | |
|------------------------------------------------|-----------|
| Linux Apache Users | 13 |
| Java Server Package | 13 |
| Web Connections and Ports | 14 |
| Apache Support and Maintenance | 14 |
| Restarting Apache | 15 |
| Diagnostic Tools for Apache | 15 |
| Admin Portal User Assistance | 15 |
| Getting Help in the Admin Portal | 15 |
| Tooltips | 15 |
| Expanding Help | 16 |
| Pipeline Pilot Help Center | 17 |
| Using the Help Center | 17 |
| Searching the Help Center | 18 |
| Chapter 2: Status and Monitoring | 21 |
| Managing Jobs | 21 |
| Jobs Overview | 21 |
| Job Management | 22 |
| Queued Jobs | 22 |
| Running Jobs | 23 |
| Job Settings | 24 |
| Job Folder Maintenance | 26 |
| Maximum Job Age Based on Job Folder Size | 27 |
| Scheduled Tasks | 27 |
| Overview | 27 |
| Scheduling Tasks | 28 |
| Monitoring scheduled tasks | 28 |
| Displaying Server Information | 29 |
| Environment | 29 |
| Ports | 29 |
| Paths | 29 |
| Monitoring Server Usage | 29 |
| Chapter 3: Reports | 31 |
| Monitoring XMLDBs with Catalog Search | 31 |
| Overview | 31 |
| Using Catalog Search | 31 |

| | |
|---------------------------------------------|-----------|
| Query Options | 33 |
| Exporting Results to Excel | 34 |
| Completed Jobs | 34 |
| Filtering the Jobs Log | 35 |
| Installed Collections | 35 |
| Installed Packages | 38 |
| Server Usage Report | 39 |
| Report Content | 39 |
| Usage Categories | 40 |
| Validation Report | 40 |
| Overview | 40 |
| Customizing the Validation Schedule | 40 |
| Manually Running a Validation | 41 |
| Validation Report Results | 41 |
| Exporting Results to Excel | 41 |
| Foundation Applications | 42 |
| Chapter 4: Security | 43 |
| Security Overview | 43 |
| Authentication | 43 |
| Support for SAML Authentication | 43 |
| Impersonation | 43 |
| Authorization | 44 |
| Pipeline Pilot System Permissions | 44 |
| Pipeline Pilot System Groups | 44 |
| Managing Pipeline Pilot Authorization | 44 |
| Authorization Guidelines | 45 |
| Access Rights | 45 |
| XMLDB Access Rights Overview | 45 |
| Tab Publishing Rights | 46 |
| Managing Access Rights | 46 |
| Impersonation and File Access Rights | 50 |
| Authorizing Client Administrators | 50 |
| Authentication | 51 |
| Authentication Overview | 51 |
| Authentication Method | 51 |

| | |
|-------------------------------------------------------------------|----|
| SAML Web SSO Settings | 57 |
| Anonymous Access | 57 |
| Support for Kerberos via SPNEGO | 57 |
| Passing Insecure Passwords | 59 |
| User Directory Sharing | 59 |
| Restricting Permissions | 59 |
| Notification Protocols | 60 |
| Security for Linux Authentication | 60 |
| Linux Shadow Password Support | 61 |
| Linux Username Case Sensitivity | 62 |
| Managing a User List for Authentication | 62 |
| Managing Impersonation | 63 |
| Windows Requirements for Impersonation | 63 |
| Linux Requirements for Impersonation | 64 |
| Client User Requirements under Impersonation | 64 |
| Enabling Impersonation | 65 |
| SAML Web SSO | 65 |
| First-time Authentication | 65 |
| Limitations | 66 |
| Configuring SAML Web SSO | 66 |
| Setting up SSO and SAML Certificates | 66 |
| Enabling SAML SSO and Configuring Service Provider Settings | 67 |
| Configuring Identity Provider Settings | 67 |
| Verifying your Configuration | 68 |
| Hub Connection | 68 |
| Groups and Permissions | 68 |
| Groups and Permissions Overview | 68 |
| System Groups | 68 |
| Pipeline Pilot System Permissions | 69 |
| Admin Portal Users | 71 |
| Custom Groups and Permissions | 71 |
| Support for Groups and Permissions in Pipeline Pilot | 71 |
| Getting Started with Groups | 72 |
| Group Management Features | 72 |
| Managing Custom Groups | 76 |

| | |
|----------------------------------------------------------|------------|
| Adding a New Custom Group | 77 |
| Removing a Custom Group | 77 |
| Renaming a Custom Group | 78 |
| Managing Group Assignments | 78 |
| Assigning Users to Groups | 78 |
| Assigning Group Member Types | 78 |
| Mapping External Claims | 79 |
| Getting Started with Permissions | 79 |
| Permission Management Features | 80 |
| Managing Custom Permissions | 83 |
| Adding a New Custom Permission | 83 |
| Removing a Custom Permission | 84 |
| Renaming a Custom Permission | 84 |
| Managing Permission Assignments | 84 |
| Certificates | 87 |
| Managing SAML Certificates | 87 |
| Overview | 87 |
| Importing Key Pairs into the SAML Stores | 88 |
| Adding Trusted Certificates | 90 |
| Managing SSL Certificates | 92 |
| SSL Certificates | 92 |
| Configuring SSL Certificates in the Admin Portal | 92 |
| Using SSL Certificates from Recognized Authorities | 93 |
| Generating a Self-Signed Certificate | 96 |
| Using the BIOVIA Self-Signed Certificate | 97 |
| Package Editors | 97 |
| JAAS Configuration | 98 |
| Pipeline Pilot JDBC Connections | 98 |
| 3D Passport Service Settings | 98 |
| Chapter 5: Setup and Configuration | 100 |
| Catalog Settings | 100 |
| Required Permissions | 100 |
| Updating Catalog Index | 100 |
| Scheduling Catalog Sync Settings | 100 |
| Updating Sync Settings | 100 |

| | |
|-----------------------------------------------------------|-----|
| Data Sources | 101 |
| Creating Data Source Connections | 101 |
| ODBC (PP) Data Sources | 102 |
| JDBC Data Sources | 102 |
| ODBC (DSN) Data Sources | 103 |
| MongoDB Data Sources | 103 |
| Testing Data Source Connections | 103 |
| Advanced Data Source Connection Configurations | 104 |
| Connection Pooling | 104 |
| Security | 105 |
| Initializing the Connection | 108 |
| Additional Information | 108 |
| Query Service Settings | 108 |
| Exporting and Importing Data Sources | 108 |
| Tagged Resources | 110 |
| Package Developer tasks | 112 |
| Creating Tagged Resource Templates | 112 |
| Creating data access components | 113 |
| Administrator Tasks | 114 |
| User Access | 114 |
| Access Rights Field | 114 |
| Package Requirement Field | 114 |
| Adding a Tagged Resource for an External Resource | 115 |
| Adding a Tagged Resource for SharePoint Online Site | 115 |
| Adding a Tagged Resource for AWS S3 Site | 115 |
| Protocol User Access to External Resources | 116 |
| File Browser Access Settings | 116 |
| File Browser Shortcuts | 116 |
| File Browser Access | 117 |
| Folder Locations | 117 |
| Managing Folder Locations | 117 |
| Overview | 117 |
| Redirecting Folders | 118 |
| Redirecting the User Folder | 119 |
| Redirecting the Jobs Folder | 119 |

| | |
|------------------------------------------------------------------|-----|
| Redirecting the Shared Public Folder | 120 |
| Redirecting the Local Temp Folder | 120 |
| Redirecting the Upload Folder | 121 |
| Redirecting the XMLDB Folder | 121 |
| Preparing to Redirect an XMLDB Folder | 121 |
| Global Properties | 122 |
| Job Settings | 123 |
| Proxy Settings | 125 |
| Setting up a Proxy Server Exclusion List | 125 |
| Server Deployments | 126 |
| Server Deployments Overview | 126 |
| Configuring Pipeline Pilot to Support Advanced Deployments | 126 |
| Reverse Proxy Deployments | 128 |
| Reverse Proxy Features | 128 |
| Guidelines | 128 |
| Configuring a Reverse Proxy | 128 |
| Load Balancing Deployments | 129 |
| Load Balancing Features | 129 |
| Guidelines | 130 |
| Configuring for Load Balancing | 130 |
| Distributed Grid Computing Deployments | 131 |
| Grid Features | 131 |
| Installation | 131 |
| Guidelines | 131 |
| Configuring for Grid Operation | 131 |
| Testing a Grid Engine with Pipeline Pilot | 133 |
| Clustering Deployments | 134 |
| Clustering Features | 134 |
| Installation | 135 |
| Clustering Modes of Operation | 135 |
| Support for Clustering | 135 |
| Guidelines | 135 |
| Configuring for Clustering | 136 |
| Server Configuration | 138 |
| Configuring Pipeline Pilot Servers | 138 |

| | |
|-------------------------------------------------------------------|------------|
| Recommendations for CPU Usage | 143 |
| Remote Administration Access | 144 |
| SSL Security Level | 144 |
| Configuring a Single Port Operation | 145 |
| Reconfiguring Ports | 145 |
| Settings for Web Client Hosts | 146 |
| Server Registry | 148 |
| Validation Rules | 148 |
| Publication Targets | 149 |
| Privacy Policy | 150 |
| Remote Administration Access | 150 |
| Chapter 6: Server Maintenance | 151 |
| Setting up MongoDB | 151 |
| MongoDB Overview | 151 |
| Requirements | 151 |
| Installing and Deploying MongoDB | 152 |
| Configuring a Pipeline Pilot Server to use a MongoDB Server | 153 |
| Configuring MongoDB as a Secure Data Source | 154 |
| Database Schema Update | 155 |
| Importing/Exporting Configurations | 155 |
| Exporting a Server Configuration | 156 |
| Importing a Server Configuration | 156 |
| Managing Licenses | 156 |
| Adding a license file | 156 |
| Removing a license file | 156 |
| Viewing license information | 156 |
| Managing the Server | 157 |
| Managing Servers | 157 |
| Restarting the Server | 157 |
| Configuring Java Servers | 157 |
| Overview | 157 |
| Java Server Configuration | 157 |
| Taking the Server Offline for Maintenance | 158 |
| Managing the Jupyter Notebook Server | 159 |
| Managing the XMLDB | 159 |

| | |
|------------------------------------------------------------------------------|------------|
| Backing Up XMLDB Files | 159 |
| Backup guidelines | 160 |
| Backing up your XMLDB | 160 |
| Purging XMLDB Files | 160 |
| How purging impacts versioning | 161 |
| Restoring XMLDB Backups | 161 |
| XMLDB versions | 161 |
| Compressing an XMLDB | 162 |
| Sharing an XMLDB with Multiple Servers | 163 |
| Chapter 7: Pipeline Pilot Services | 165 |
| Managing Pipeline Pilot Services | 165 |
| Using the Windows Services Console to Manage Pipeline Pilot Services | 165 |
| Managing Pipeline Pilot Servers on Linux | 166 |
| Shutting Down Pipeline Pilot on Linux | 166 |
| Manually Starting the Linux Server | 167 |
| Pipeline Pilot Manager | 167 |
| Managing Services with Pipeline Pilot | 167 |
| Overview | 167 |
| Pipeline Pilot Service Tasks | 167 |
| Platform Builder Task | 168 |
| Version Change Process | 169 |
| Running Pipeline Pilot Services in Console Mode | 169 |
| Console Mode Commands | 170 |
| Appendix A: Support for Security Issues | 171 |
| Logon Support | 171 |
| Resolving Administrator Lockout Problems | 171 |
| Deleting AuthConfig.xml and Logging on using scitegicadmin Credentials | 171 |
| Deleting Authusers.xml to Reinstall the scitegicadmin Account | 171 |
| Backing up {root}/xmlpdb/Object files One-by-One | 172 |
| Recommended Additional Security Settings | 172 |
| Appendix B: Client-Server Deployment Issues | 173 |
| Overview of Grid Job Processing | 173 |
| Guidelines for Grid-friendly Protocols | 173 |
| Running Protocols on a Grid | 174 |
| Running a Complex Job on Multiple Nodes | 174 |

| | |
|-------------------------------------------------------------------|------------|
| Adding Run on Grid Parameters to Legacy Protocols | 174 |
| Batch Size Recommendations for Parallel Protocol Processing | 175 |
| Clustering Scenario | 175 |
| Grid Scenario | 176 |
| Fine-tuning Clustering Configurations | 176 |
| Enabling Clusters used as Endpoints | 176 |
| Cluster Job Handling | 177 |
| Job Folder Maintenance | 178 |
| Job Folder Lifetimes | 178 |
| Maximum Job Age Based on Job Folder Size | 179 |

Chapter 1:

Introduction

Pipeline Pilot Server Overview

The BIOVIA Pipeline Pilot Server is an application server that streamlines the integration and analysis of vast amounts of scientific data. Pipeline Pilot makes it easy to create scientific web services that can be used independently or as part of a company's Service Oriented Architecture (SOA) strategy. Pipeline Pilot provides an agile development environment, fast and secure deployment, minimal maintenance costs, and application extensibility.

Supported Operating Systems

Pipeline Pilot is supported on Microsoft Windows and various Linux distributions. For details about currently supported operating system versions, see the *Pipeline Pilot System Requirements Guide*.

Client Software

Pipeline Pilot can be accessed in many ways. For the computational expert, the Pipeline Pilot Client provides a rich authoring environment for protocol design and execution. Protocols can be easily published for use in an intuitive web browser interface via Pipeline Pilot Web Port.

Users can run client software to create, edit, and view protocols on their local client machines. They publish files (protocols and components) on the server and run their protocols on the server as well. To run a protocol, the client software launches a "protocol job" on the server. The server monitors the job for status and other data, so it knows when the job is completed and when the job requires some specific client-side interaction.

Developer Tools

A rich set of Software Development Kits (SDKs) is provided that can be used to integrate Pipeline Pilot with other solutions. Additionally, Pipeline Pilot protocols can be easily published as web services for use in a Services Oriented Architecture (SOA). Both SOAP and RESTful web services can be deployed.

Enterprise Architecture

Pipeline Pilot employs a services-oriented approach that lowers the cost of integration into an enterprise environment. This architecture provides a convenient way to interconnect the platform and distribute resources efficiently across different locations.

Pipeline Pilot services are supported on single servers, load-balanced servers, and grids/clusters (Linux only). Pipeline Pilot is commonly deployed within large networks with central database management systems, numerous client workstations, and at local and remote sites.

Pipeline Pilot employs a web services architecture to support modular communication between the server and clients. The server handles SOAP and other HTTP requests from clients, managing and balancing numerous simultaneous requests.

Protocol Database Sharing and Data Access

By default, Pipeline Pilot stores its database of components and protocols locally. Pipeline Pilot can also be configured to share this database to facilitate re-use of protocols across multiple sites and/or servers in an organization.

Pipeline Pilot can access data that resides on the file server (files or databases). It can also access database servers elsewhere on the network and files on network-shared disk drives via their Uniform Naming Convention (UNC) path name, (for example, "\\server3\data\file2.sd").

Data sources and ODBC Data Source Names (DSNs) need to be defined on the server, not on the client, so components can use them. Because the protocol runs on the server, it cannot access files that reside on client machines in areas of the file system that are not shared. For protocols to access the data correctly, these files need to be moved to shared disks.

Security

Pipeline Pilot includes extended security capabilities to support authentication, impersonation, and authorization.

- **Authentication:** Confirms network user identities to control server access. Supported authentication methods include file-based, local, and Windows domain. Additional features supported with authentication include anonymous access, user directory sharing, Kerberos (via SPNEGO), SAML (Sender Vouches), and insecure basic authentication.
- **Impersonation:** Supported on both Windows and Linux, allows running protocol jobs to access required server resources and controls permissions to files on the server.
- **Authorization:** Defines client user permissions for performing tasks.

Java Server Package

Pipeline Pilot includes an optional Java Server package that installs Apache Tomcat Java web server. It is only necessary to configure this server if you are installing a package that requires it.

About this Guide

This guide is for system administrators and IT professionals. It provides essential information for managing Pipeline Pilot on your servers and for supporting client users on your network. This guide assumes you are familiar with server administration on Windows and Linux platforms and that you have some experience with BIOVIA client applications (e.g., Pipeline Pilot Client, Pipeline Pilot Web Port, Discovery Studio, etc.).

Getting Additional Information

The following documents are included with your Pipeline Pilot documentation zip file included with the installation files:

- Pipeline Pilot Server Installation Guide
- Pipeline Pilot System Requirements
- R Software Installation and Configuration Guide

Notes:

- The *Pipeline Pilot Server Installation Guide* provides all the information you need to get your server up and running and available for initial use. It also provides important post-installation tasks you may need to perform, to ensure that your server is properly configured for your type of operating system and deployment.
- Instructions for all post-installation tasks handled in the Pipeline Pilot Admin Portal are covered in this guide or in other support and configuration guides available with separately licensed collections.
- Clients can access the Pipeline Pilot Server Home Page and install/upgrade Pipeline Pilot Client. A *Pipeline Pilot Client Installation Guide* is available on the home page that provides instructions. For additional information, see [Client Support](#).

Additional Information

For more information about Pipeline Pilot and other BIOVIA software products, visit BIOVIA Support on the Web:

[Dassault Systèmes User Communities](#)

Getting Started with BIOVIA Pipeline Pilot Server

System administrators have efficient ways to handle the high volume of users, multiple installations, multiple sites, managing shared resources between users, scalability, security, administration tools, and portability. Your administration tools include:

- [Server Home Page](#): A web page for accessing Pipeline Pilot Servers and related applications. It opens in a browser window and provides a way to access programs, documentation libraries, and get assistance.
- [Administration Portal](#): After Pipeline Pilot is installed, all required server settings are saved in XML files. If you need to change any configurations after installation, change the server settings directly in the Admin Portal. This web-based tool provides secure access from any network location, independent of the server location.
- [Pipeline Pilot Manager](#): Tools for managing Pipeline Pilot services.

Administration Portal Overview

System administration 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, Firefox, and Chrome.

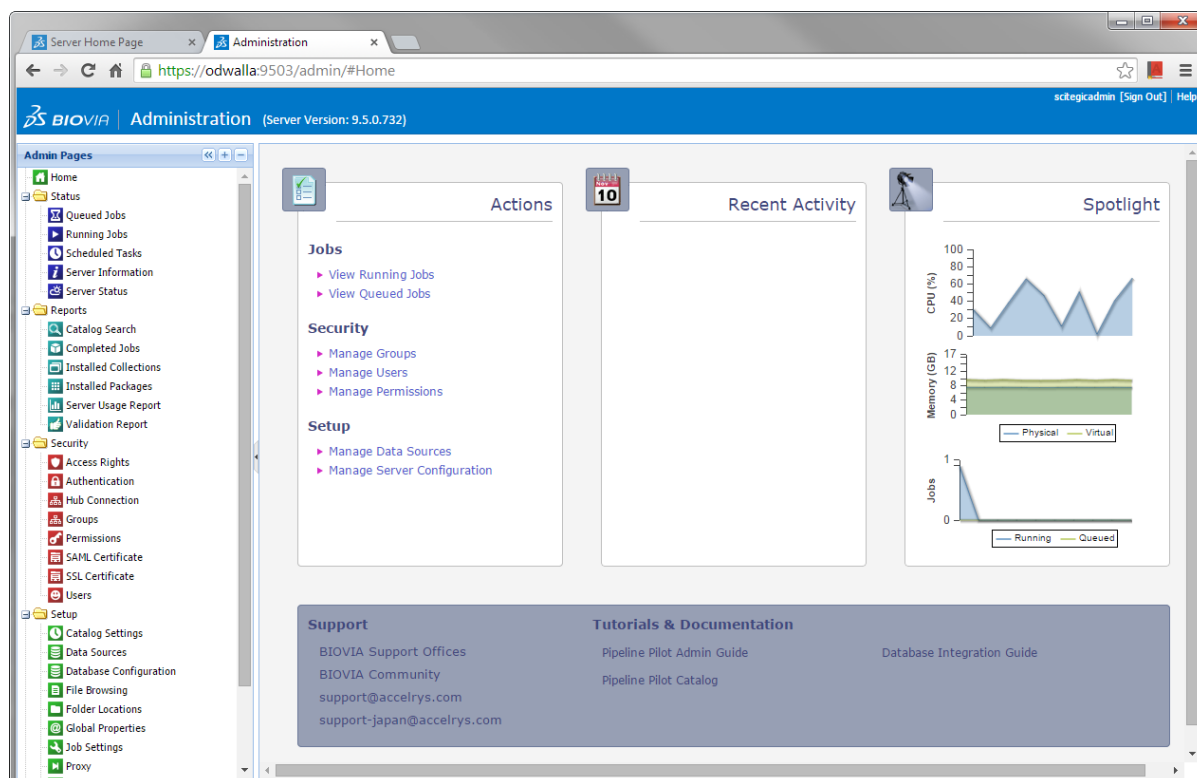
Use the Admin Portal to manage the server, control access to the protocol and component database and maintain the XMLDB files (backup, restore, purge operations), handle security (authentication and impersonation), and cancel jobs.

Accessing the Admin Portal

To access the Admin Portal, you need a valid user name and password combination. An administrator only needs Pipeline Pilot administrative authority, independent of local or domain users. The default settings are: "scitegicadmin" (user name) and "scitegic" (password).

To log onto the Administration Portal:

1. From the [Home Page](#), select **Administration Portal**.
2. Enter the default user name and password and click **Sign In**.



Note: If remote administration is disabled, you can only access the Admin Portal on the local server. By default, remote access is enabled when Pipeline Pilot is installed on your server. (For further details, see [Configuring Pipeline Pilot Servers](#)).

Admin Portal Home Page Features

The Admin Portal Home Page provides quick links to the most commonly required features, your most recently accessed features, and an overview of your Pipeline Pilot Server's current status. These links include:

- **Actions:** Directly links to the features providing the most commonly required information – Jobs, Security, and Setup.
- **Recent Activity:** Provides links to your 10-most-recently-visited features.
- **Spotlight:** Summarizes the current CPU and memory usage for your Pipeline Pilot Server and the number of jobs currently running and queued.
- **Support:** Access BIOVIA support and Community or send emails to support.
- **Tutorials & Documentation:** Access guides to help you configure your Pipeline Pilot Server for your requirements.





Admin Pages Explorer

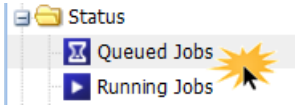


The Admin Portal is navigated using the Admin Pages Explorer. This organizes the pages of the Administration Portal into folders and allows you to navigate directly back to the *Home* page.

| Folder | Purpose |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Maintenance | Allows you to manage your server, prevent access during maintenance, back up, clean, or restore your server, and import or export the server configuration to facilitate consistent set up of more than one server. |
| Reports | Provides information on the collections and packages available on your server, reports on usage and validity of the installed components and protocols, and a history of completed jobs run on your server. |
| Security | Allows you to employ a wide range of security settings to control authentication, certificates, access, and to configure groups, permissions, and users. For further details, see Security Overview . |
| Setup | Allows you to perform detailed configuration of your server; including validation rule, file, and folder locations, proxy settings, connections to databases, job and catalog settings, global properties, server configuration, server registry, and clustering and grid configuration. |
| Status | Provides monitoring facilities for queued and running jobs and the server's current status, task scheduling, and a summary of the server information. |
| Custom folders | Some applications or collections may have their own custom administration settings pages. For more information on configuring these settings please refer to the documentation for the relevant application or collection. |

Displaying the Admin Pages Explorer

The Admin Pages Explorer is located on the left side and offers an expanding/collapsing hierarchy of folders to the Admin Portal features. Depending on how you like to work, it can be hidden, resized, and have all its folders expanded or collapsed.

| To: | Use these features: |
|-------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| Hide the Admin Pages Explorer tree |  Hide |
| Show a previously hidden Admin Pages Explorer tree  |  Show |
| Expand all folders and display all features in the tree hierarchy |  Expand tree |

| To: | Use these features: |
|------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Open an Admin Portal feature | Click an icon or the text for the feature you want to use.  |
| Collapse all folders and hide all features in the tree hierarchy |  Collapse tree |
| Resize the width of the Admin Pages Explorer pane | However over the blue lines at the right-hand border until your cursor changes into a resizer arrow handle. Then drag the border left (to narrow the width) or right (to widen).  |

Client Support

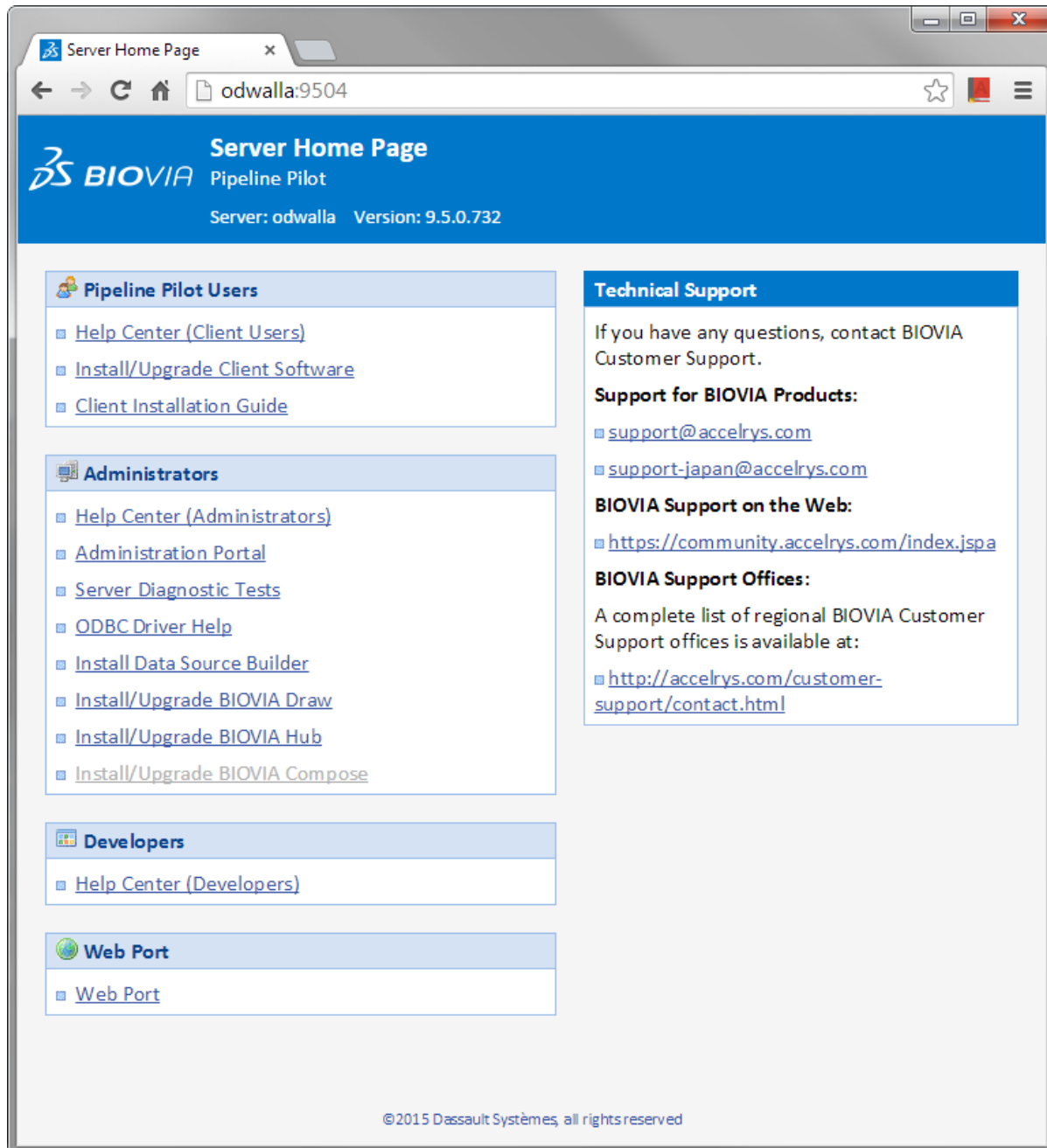
Overview of Client Types

Client products provided with Pipeline Pilot include Pipeline Pilot Client and Pipeline Pilot Web Port (Web Port). All client software runs on Windows desktops.

- **Pipeline Pilot Client:** Appropriate for users who need to design protocols and publish them to shared tabs for reuse with others. Also includes features for integrating the client software with other applications.
- **Web Port:** A web-based user interface to a Pipeline Pilot Server. It serves as a template and as an example of how to use Pipeline Pilot as a web service with your own client applications. Anyone who does not have Pipeline Pilot Client installed can run protocols from Web Port via their web browser.

Client Installation Overview

Pipeline Pilot Client installation is web-based. From a browser, users can install the client software from the Server Home Page:



Automatic Client Software Updates

Clients are programmed to detect when they require updates if they are connected to a server that is incompatible in some way. The situation is explained to the user and if requested, the client software downloads a new client installation and initiates a new client installation. (If required, multiple client installations may exist on a client machine, but they must be installed in different locations.)

Tip: You can disable automatic client download on the [Setup > Server Configuration](#) page of the Admin Portal.

To update a client:

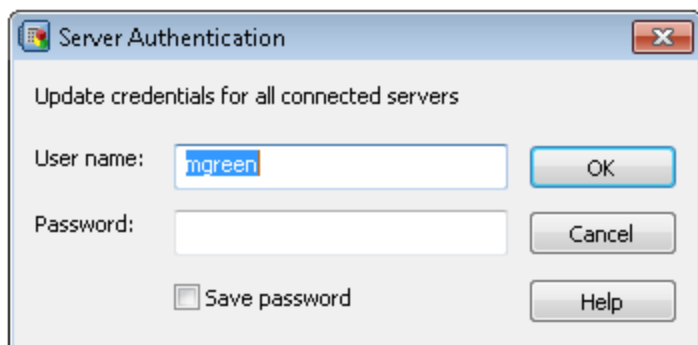
1. Open **Pipeline Pilot Client** and change the active server location.
If the server you are connecting to has a later version of Pipeline Pilot available, this incompatibility will be detected.
2. Click the **Download** button on the incompatibility warning dialog.
The client installer will automatically be downloaded and started.
3. Install Pipeline Pilot using the wizard provided, ensuring that you install to a new location if you want to maintain access to your previous version.

Confirming User Identity

You can restrict access to Pipeline Pilot based on the identity of your network users. The method employed to verify users is based on how a server is configured to handle authentication. It confirms the identities of your users. When a user starts a Pipeline Pilot Client, the Windows username identifies the user.

If Pipeline Pilot is configured to use authentication, users are required to enter a password each time they log into Pipeline Pilot. Their logon information is validated. Names and passwords are compared against an authorized list. If the system detects a match, access is granted to the extent specified in the permission list for each user.

A feature is available in Pipeline Pilot to update credentials for all server connections (**Tools > Options > Change User Identification**).



Note: If any users with client impersonation run protocols as scheduled tasks, they must select to save their passwords in this dialog. This way, the batch files can run automatically at night or on weekends without interrupting the job processing for passwords.

BIOVIA Desktop Connector

The BIOVIA Desktop Connector enables communication between BIOVIA applications running in your browser and desktop applications, such as BIOVIA Draw, ChemDraw, or Microsoft Office. It replaces the BIOVIA Plugin, which was included in previous Pipeline Pilot releases.

IMPORTANT! If you are using the Plugin, you must upgrade to the Desktop Connector.

The Desktop Connector runs in two modes:

- **Messaging Service (Recommended):** This mode offers improved security and support for the Microsoft Edge browser. It requires a Windows server in order to operate.
- **Legacy:** This mode is identical in function to the Plugin, and can operate on Windows and Linux servers.

IMPORTANT! Legacy mode is supported only until September 4, 2021, when the current security certificates expire. If you are using Legacy mode, you must upgrade to Messaging Service mode.

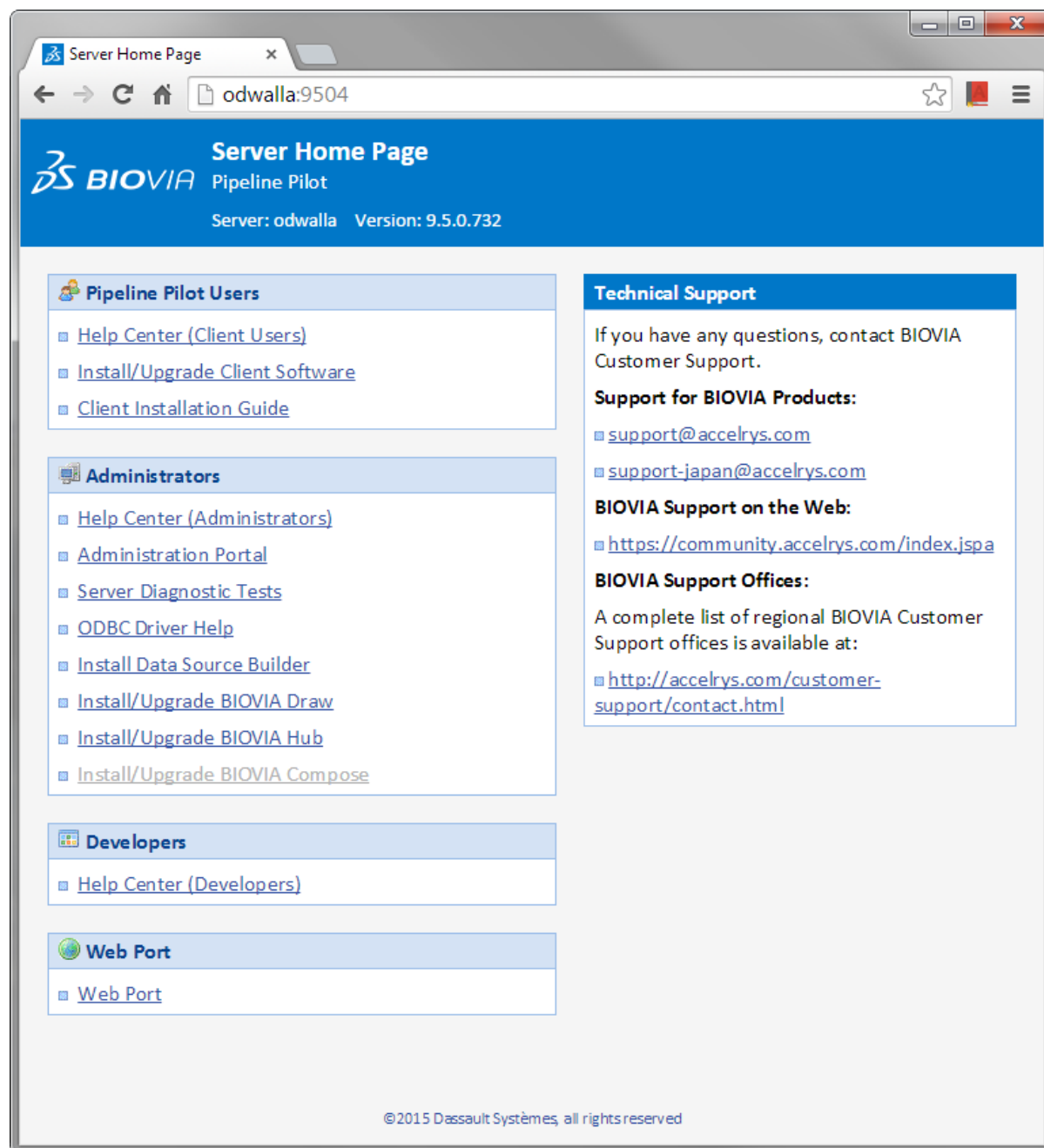
For information on how to configure the Desktop Connector and the global properties for a chemistry sketcher integration, log in to the Pipeline Pilot Server Home Page, and then click **Set Up BIOVIA Desktop Connector**.

After you set up the Desktop Connector in Messaging Service mode, users connecting to a Pipeline Pilot-based web client can download and install the Desktop Connector client, which connects their browsers to their desktop applications.

Note: Users must install and run the Desktop Connector client with ordinary user privileges, not as an administrator of their own machine.

Server Home Page

The Server Home Page is a one-stop place for all things related to interface to a Pipeline Pilot Server applications. It opens in a browser window and provides a way to get assistance and access documentation and programs. From the home page, you can install or upgrade a Pipeline Pilot Server, access the Help Center, run the Admin Portal, access RSS feeds, and launch Web Port.



Accessing the Home Page

You can access the Server Home Page in the following ways:

| From Here: | Do this: |
|-------------------------------------------|---------------------------------------|
| Web browser on your Pipeline Pilot Server | Use this URL: "http://localhost:9944" |

| From Here: | Do this: |
|-----------------------------|----------------------------------------------------------------------------------------------------------|
| Web browser on any computer | Use this URL: "http://<servername>:9944" Where <servername> is the name of your Pipeline Pilot Server |
| Pipeline Pilot Help menu | Select Server Home Page |

Notes:

- If your server runs on a different port, substitute your port number for "9944" (the default).
- Help Center offers documentation specifically tailored to system administrators, developers, and Pipeline Pilot users. You can access it from the Server Home Page.

Managing Administrator Access

After you initially log onto the Admin Portal using the default settings, we recommend that you change the username and password so that the default account cannot be accessed by users who should not have Admin Portal Access.

To change the password for the default Admin Portal user:

1. Go to **Security > Users**.
2. Select **scitegicadmin** from the **Users** list.
3. Enter the new **Password** and then re-enter it for **Confirm Password**.
4. Click **Add**.

You should ensure that the correct users can access the Admin Portal, this can be controlled in a number of ways:

- membership of the *Platform/Administrators* group
- membership of any group which has the *Platform | Administration | Logon* permission
- direct assignment of the *Platform | Administration | Logon* permission to a user

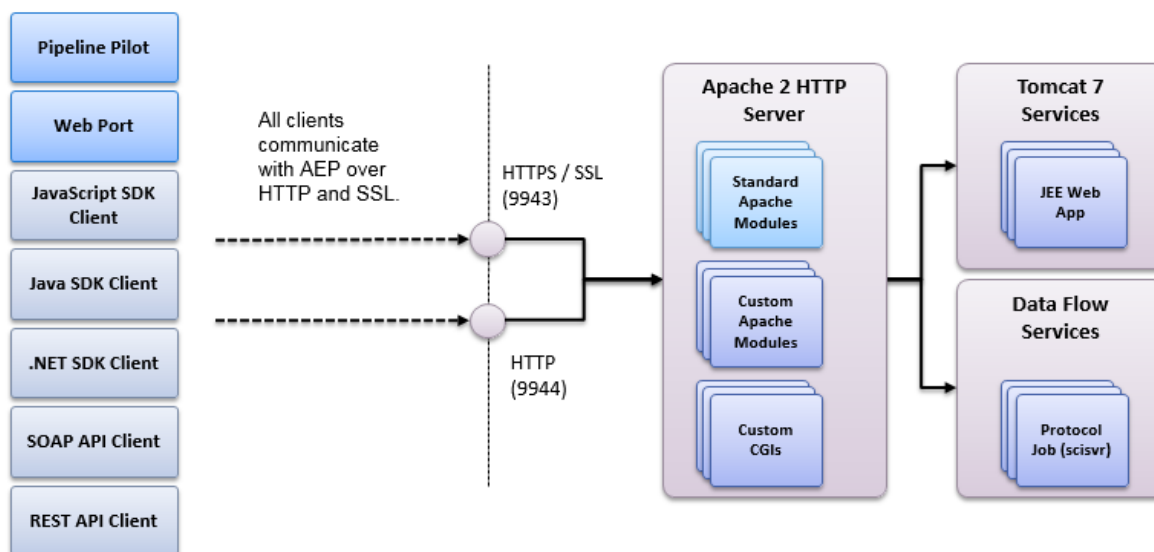
You can explicitly prevent users or groups from accessing the Admin Portal by adding the users or groups to the *Denied Users* or *Denied Groups* list for the *Platform | Administration | Logon* permission.

Tip: Refer to the [Managing Permission Assignments](#) and [Managing Group Assignments](#) topics for further instructions on assigning permissions to users and groups.

Whenever you log into the Admin Portal, your browser creates a temporary cookie that stores the username and password of the user currently logged onto the system. Only the server can read this cookie. This temporary file is deleted after you close the browser or log out of the portal.

Web Services Overview

Pipeline Pilot provides several ways to publish and access HTTP-based web services. Both SOAP and RESTful web services can be accessed through various client channels. For example, Pipeline Pilot provides language-specific client SDK's, protocol-specific SOAP services and RESTful web services.



Apache 2 HTTP server as the communication gateway to all Pipeline Pilot web services

Client SDK Access

Pipeline Pilot SDK's that make it easy for you to develop custom clients in popular environments such as JavaScript, Java and .NET. All client SDK's use the same internal SOAP API's as Pipeline Pilot and Web Port. However, the client SDK's enable you to write code against a stable interface and abstract your code from the details of calling the server.

SOAP Web Services API

Pipeline Pilot exposes a SOAP-based web services API accessible to client applications. Any client application can talk directly to the protocol and folder-based SOAP services in lieu of using a client SDK. SOAP-based protocol services are a good choice when you want to provide a more fixed web services interface.

RESTful Web Services API

As of Pipeline Pilot 9.0, a new RESTful web services API is available. This API allows client applications, particularly modern AJAX-based web applications, to call the server and launch protocol jobs. Package developers can create RESTful URL routes to protocol-based services using the URL routing configuration. For further information, see the *RESTful Web Services Guide* (go to Help Center Developers tab > Client-side Integration).

Apache 2 HTTP Server

Pipeline Pilot uses HTTP-based web services to provide modular, portable communication between the server and the various client types. These web services run on an Apache 2 HTTP server, supported on both Windows and Linux platforms.

Apache 2 HTTP Server (Apache) acts as a gateway to all back-end web services. Many web service are deployed directly within Apache 2's module-based system. Module-based services support requests such as user logins, locating all available web services, navigating file systems on the server, accessing the XMLDB and running protocols. However, Apache also acts as a front-end web proxy to service hosted in other containers like CGIs, JEE web applications and even protocol-based services running

within data flow servers. The value of Apache is that it provides a single point of access and control for all incoming requests.

For each client request, Apache handles ensuring that the client user is authenticated and authorized to access the back-end service or resource. Apache then routes incoming requests to the correct service container and balances these resources across all client connections. In fact, Apache can balance requests across multiple server nodes in some load balanced and clustered configurations.

Keep the following in mind when working with Apache web server software:

- Apache software is included with your Pipeline Pilot installation. All required files are automatically installed and configured at the same time you install Pipeline Pilot.
- No post-configuration of Apache is required to run Pipeline Pilot.
- Pipeline Pilot's Apache service on Windows has its own unique name and port. If your server computer is already running another installation of Apache web server software in another directory, it will not impact the Apache files installed with Pipeline Pilot.
- Do not download and install any Apache updates in the folder where Pipeline Pilot is installed. BIOVIA will provide you with any required Apache updates as they become available.

IMPORTANT! If you customize any settings related to Apache web server software in the Pipeline Pilot installation directory (not recommended), it could interfere with server performance. Any changes you make will be overwritten the next time you reinstall or upgrade Pipeline Pilot.

Windows Apache Users

On Windows, Apache is installed as a Windows service that restarts automatically upon reboot. Windows services run, by default, as the local system user who has limited network and domain access rights (e.g., the local system user cannot access UNC paths on many networks). Although this setup is appropriate for many services, we recommend that you run the service under another user who has administrative rights on the server machine. If you are not planning to use impersonation for running protocols, ensure that the Apache user has appropriate network rights for carrying out all the tasks that users' protocols require.

Linux Apache Users

On Linux, Apache is optionally configured to start by a boot script that is executed when the machine reboots. If you are not using impersonation, all resource access rights for running protocols are based on the Apache user account specified during installation. Ensure that the Apache user has appropriate rights for carrying out all the tasks that the users' protocols require (such as access to mounted drives).

Java, .NET, and JavaScript can talk to dedicated APIs, provided as part of the client side SDKs, which abstract the internal web services layer.

Java Server Package

Pipeline Pilot includes an embedded Apache Tomcat server (<http://tomcat.apache.org/>). Tomcat must be installed and running to support core services such as task scheduling, fast chart updates, and Query Service. The Pipeline Pilot installer automatically configures and starts a Tomcat service for you.

Note: Pipeline Pilot is tuned and tested to run only the JEE web applications included in the Pipeline Pilot installer. Do not deploy additional web applications to the embedded Tomcat as doing so may make the server less stable.

Web Connections and Ports

The Apache web server supports transmission security and uses both HTTP and HTTPS network protocols. HTTP Data transfers are made as clear text. HTTPS is an encrypted form of HTTP, implemented through the Secure Sockets Layer (SSL) standard. The encrypted secure connection is created by running an ordinary HTTP connection on top of an encrypted SSL connection. Using HTTPS in the URL directs the message to a secure port number, so the session can be managed by a security protocol (SSL) that encrypts all transmitted messages for online transmission.

SSL is the leading security protocol on the web. In addition to protecting user login information, an SSL session is used when system administrators work with the Admin Portal.

The default ports for Pipeline Pilot on the Apache web server are:

| Port | Type | Description | Example |
|------|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| 9944 | HTTP | The HTTP port is the primary server port. It is always in use and supports Pipeline Pilot Web Port, the Pipeline Pilot Help Center, and all non-secure web services. | http://<server name>:9944 Server Home Page URL, where <server name> is the name of your server |
| 9943 | HTTPS | The HTTPS port is managed by a security protocol (SSL). It is always in use and supports the Admin Portal and the locator web service that includes the user login procedure. | https://<server name>:9943/admin Administration Portal URL, where <server name> is the name of your server. |
| 9945 | Tomcat Shutdown Port | | |
| 9946 | Tomcat HTTP Port | | |
| 9947 | Derby port | | |

Notes:

- You can use other port numbers, if the defaults described above are used by other services on your server system or by previous installations of Pipeline Pilot.
- You can also configure your server for single port operation (primarily applies to standalone operations).

Apache Support and Maintenance

The Apache web server is installed when you install Pipeline Pilot. You do not need to manage the Apache web server separately. All required files for Apache are installed at the time you install Pipeline Pilot and are updated whenever you upgrade to a newer version. For troubleshooting purposes on your network, you can use the Admin Portal to look up process information about your Apache web server. This is especially useful if you have multiple instances of Apache and Pipeline Pilot running at the same time.

To look up process information for your Apache web server:

- Select **Status > Server Information**. A list of information related to the current Apache web service is displayed. See [Displaying Server Information](#) for further information.

Restarting Apache

You only need to restart Apache if you made some changes on your server that require a restart. You can perform this type of restart from the Admin Portal. Before you begin, ensure that there are no jobs currently running on your server, Apache cannot restart if protocol jobs are active.

To restart Apache in the Administration Portal:

1. Confirm that no jobs are running on the same server (**Status > Running Jobs**).
2. Select **Maintenance > Manage Server**.
3. Click **Restart Apache**.

Diagnostic Tools for Apache

The Administration Portal includes tools to help you test your Apache server. The **Server Information** page includes the following features:

- **Apache Server Status:** Details of your Apache server's performance. The statistics indicate workload organized across the threads and processes (depending on OS platform) and reveal information about recent requests. You can adjust the refresh rate on this page.
- **Apache Server Information:** Detailed set of configuration data about your Apache HTTPD server and its installed modules. This offers another way to look at data that is accessible in the Apache "httpd.conf" file and the various package-defined extensions in the server configuration.
- **Tomcat System Properties:** If a Java server is installed and running information related to the Apache Tomcat server is provided. For example, you can get information about data on ports, file paths, and versions. Information is also displayed if a Java server is not installed or is currently not running.

Admin Portal User Assistance

The Admin Portal provides assistance in a number of ways:

- Tooltips
- Expanding help and dynamic help
- Pipeline Pilot Help Center

Getting Help in the Admin Portal

The Admin Portal includes many helpful tips directly on the page where you are making changes.

Tooltips


With some options, tooltips are available. To access this additional help, point to a setting.


| Edit Data Source | |
|-------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Name: | GEMSTest |
| Description: | |
| Type: | JDBC |
| Access Privileges: | <input type="checkbox"/> |
| Driver: | org.apache.derby.jdbc.ClientDriver40 <input type="button" value="Import JDBC Driver"/> |
| Connection | jdbc:derby://localhost:9997/GEMSTest |
| Connection Timeout: | |
| Optional DB Username: | scitegicadmin |
| Optional DB Password: | ••••• |
| Advanced Settings: <input type="checkbox"/> | |
| Query Service Settings: <input type="checkbox"/> | |
| <input type="button" value="Save"/> <input type="button" value="Test"/> | |

Specify the complete JDBC connection string required by the driver. Refer to the Database Integration Guide for more information.

Expanding Help

With some options, expanding help is available. To access this additional help, point and click on a setting.

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| XMLDB Endpoint | %httproot%/scitegic/xmldb |
|  The SOAP URL of the AEP web service for access to the component and protocol XML database. This may reference a service on this or another server; the latter facilitates the sharing of an XML database across AEP installations. The default for a standalone server is to use the XML database service on that server. The setting for a clustered environment must reference the XML database service on the primary node of the cluster. IMPORTANT: Please restart the Apache server if you make any change to this setting to clear out all the cached data. | |
| Automatic Client Download | <input checked="" type="radio"/> Yes <input type="radio"/> No |
| File Browser | <input type="checkbox"/> Restricted |

In some instances, a help icon is available that displays expanded help in a window. Point and click on a setting, then click the  icon.

Authentication

Authentication Method

Check users against: ☒ A list of users (1 defined) ☐ An external user directory

Domain

Settings:

| | | | |
|-------------------------------|---------|----------------------------------------------|-------------------------------------|
| Default Domain(s) | ACCELRY | Allow Domain access only from listed domains | <input checked="" type="checkbox"/> |
| Allow SPNEGO (Kerberos) | | | <input type="checkbox"/> |
| Accept passwords via SSL only | | | <input checked="" type="checkbox"/> |
| Impersonation | | None | <input type="text"/> |

Anonymous Access

Username

Password



When a user name is passed to the server for authentication under Domain authentication, it can include a domain name prefix to indicate the domain directory against which to check the user name. When no domain name is included, the name is checked against the one or more domain names specified in this list. The domains are checked in the order listed. However, subsequent authentication attempts will take into account the domain where the user name was authenticated most recently.

Note: Separate multiple domain names with a comma.

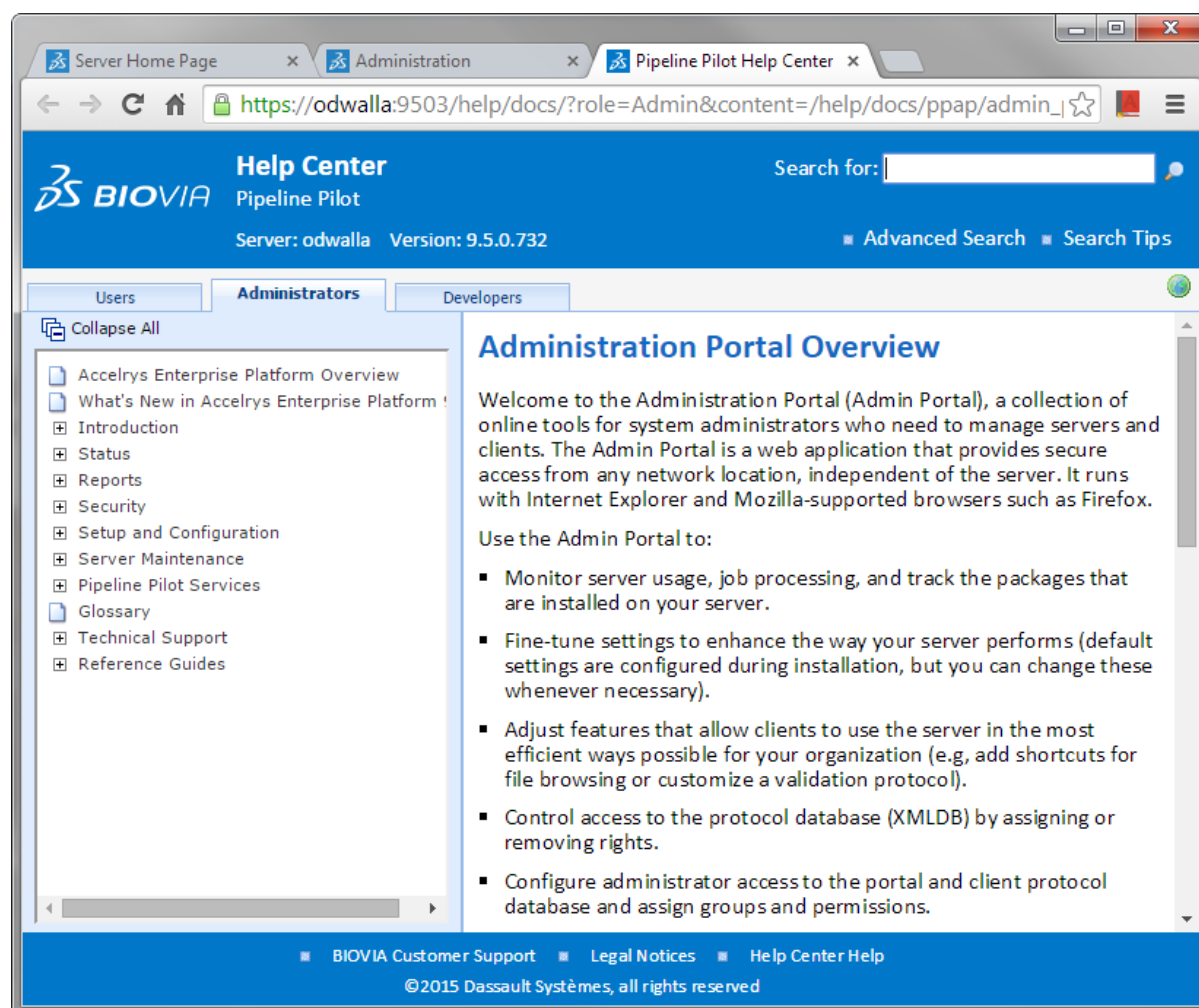
Pipeline Pilot Help Center

The Admin Portal also provides access to the Pipeline Pilot Help Center where you can find more detailed help topics for performing specific tasks.

You can get this online assistance by clicking the Help button on the upper-right corner of the Server Admin page.

Using the Help Center

Use the controls on the left to navigate through the contents tree. Use the text field in the top right to find topics that match your search criteria. View the topic content on the right side. Tabs provide access to online resources for different audiences (Users, Administrators, and Developers).



Searching the Help Center

Use the **Search for** feature in the Help Center to quickly find information. You can specify word combinations, word matching, and select document categories to include/exclude from search results (user guides, component reference help, etc.)

To use Advanced Search in the Help Center:

1. Click the **Advanced Search** link in the Help Center.
2. Specify the terms to search for in **Look for the words**.
3. Select how to combine the words in your search term for **Show search results** and how to **Match with** word variations.

4. Choose the **Search Categories** to query.

Advanced Search

| | |
|-----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Text to find in your chosen search categories | Look for the words: <input type="text" value="authentication"/> <input type="button" value="Search"/> |
| Select word combinations to include in your search | Show search results: <input type="text" value="with at least one of the words"/> ▼ |
| Word variations to include in the results | Match with: <input type="radio"/> entire words <input checked="" type="radio"/> the beginning of words |
| Expand search results to include different document types | Search Categories: <input checked="" type="checkbox"/> Pipeline Pilot Client Help (Users) <input checked="" type="checkbox"/> PilotScript Help (Functions) <input checked="" type="checkbox"/> Component Help (Collections) <input checked="" type="checkbox"/> Collection Guides (All) <input checked="" type="checkbox"/> Component Help (User-published) <input checked="" type="checkbox"/> Admin Portal Help (Administrators) <input checked="" type="checkbox"/> Integration and Development Guides, APIs (Developers) |

5. For help searching, click the **Search Tips** link.
6. Click **Search**.
7. In the results, filter the list according to the category.

Your search for **authentication** has found the following 142 document matches:

| Document | Size | All Categories ▾ | Hits |
|---------------------------------------------------|------|-------------------------------------|------|
| Changing Authentication Passwords | 1K | User | 5 |
| Changing User Logins | 1K | User | 3 |
| Copy File | 13K | Component Help: Accelrys/Generic | 24 |
| Managing Authentication | 35K | Admin | 59 |
| Move File | 13K | Component Help: Accelrys/Generic | 21 |
| Delete File | 7K | Component Help: Accelrys/Generic | 11 |
| Security Overview | 11K | Admin | 15 |
| Security for Linux Authentication | 10K | Admin | 12 |

Chapter 2:

Status and Monitoring

Managing Jobs

Jobs Overview

When a client requests to run a protocol, the running protocol and its resulting data are known as a "job". The same protocol may be used to run multiple jobs. Below are a few basic concepts related to job management.

| Term | Description |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Job name | A job name is a unique identifier. This is a 36- or 38-character identifier, such as "{A49B9BE2-62EF-4937-8AA6-3E8A960388DD}" or "ppc3DDE7-B90D-C943-EE8C-0A8701A45278". (The surrounding brace characters became optional starting with Pipeline Pilot 9.0.) |
| Job folder | A job is identified by a job folder. This is created in a user-specific folder in the Jobs directory location defined in the Admin Portal under Setup > Folder Locations (default is "<install>/web/jobs/<username>"). The folder contains files used to manage the running job in addition to files written by the protocol itself. |
| Job lifetime | <p>A job folder is created when a job is launched. Once it is running, the protocol job may write files to the job folder and it may interact with the client as dictated by its client-side components. Any connected client can see the job status by polling, up to the point when the job finishes or is canceled. The job may survive as a static folder of data for some time after the protocol run completes. During this time, clients can browse the job result data or load it into a viewer. At some point, the client requests the destruction of the job, which removes the job folder and all its data files.</p> <div>Notes:<ul style="list-style-type: none">■ The default maximum number of jobs each client user is allowed to maintain on the server is 100. Once the number exceeds this limit, older jobs are deleted. To change the maximum number of archived jobs per user, go to Setup > Server Configuration.■ Web Port does not remove client jobs. The Web Port help recommends that users remove older files no longer needed from the Jobs tab.</div> |
| Job process | <p>Each running job is represented by a single process named "scisvr.exe", spawned by the server. On Linux, the process name does not contain the ".exe" extension. There will also be processes named "scisvr" that are not running jobs (launcher daemons).</p> <div>Note: Not every "scisvr.exe" process represents a running job. For details on how job processes are created and managed, see Queued Jobs.</div> |

| Term | Description |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Result files | <p>Some protocol results consist only of data values, but many results also include files. Result files may be written to a permanent location (such as the user's data folder or some other specific storage location on the network). Alternatively, the result files may be written to the job folder itself. In this case, the results are deleted when the job is removed from the system.</p> <p>Note: When a job is started from a client, it can also be stopped from the Jobs tab in the client application. Other client programs generally have a similar explicit way of requesting job cancellation.</p> |
| Temporary files | <p>Protocols often create temporary files that are stored on the server in a directory associated with the job that creates them. Temporary files provide a convenient way to cache data during a job, and for passing data between pipelines within a protocol. In addition, some client-side programs such as Microsoft Excel can view these temporary files to display results, by referencing them as HTTP-based URLs. Other client-side programs make downloaded copies on the client machine. These temporary files survive until the job is deleted, which also deletes the temporary files.</p> |
| Queued jobs | <p>To help prevent server overload, an option is available that limits the number of jobs running on server at the same time. By default, this limit is set to 10, and you can increase or decrease the number of simultaneous jobs you want your server to support. Any excess jobs are placed in a queue and processed in order of submission, as job running slots become available. For details, see Job Settings.</p> |
| Scheduling jobs | <p>You can schedule protocol jobs to run at specific times by scheduling the invocation of the command-line application called "RunProtocol.exe". You can run this program on any machine that has Pipeline Pilot installed on it. RunProtocol is also a simple way to integrate the running of protocols from other environments such as Perl. For further details, see the RunProtocol.exe Command Line Guide (Help Center > Developers tab > Client-side Integration).</p> <p>Note: More sophisticated approaches are also available using the client-side SDKs.</p> |

Job Management

| To do this: | Go here: | See also: |
|----------------------------------------------------|--------------------------|--------------------------------|
| Customize how you want jobs to run on your servers | Setup > Job settings | Job Settings |
| View or cancel running jobs | Status > Running Jobs | Running Jobs |
| Identify jobs waiting to run | Status > Queued Jobs | Queued Jobs |
| View a list of completed jobs | Reports > Completed Jobs | Completed Jobs |

Queued Jobs

When a client runs a protocol, the running protocol and its resulting data are known as a "job". The same protocol may be used to run multiple jobs. The server has a Running Job Limit feature that specifies the maximum number of polling jobs that can simultaneously run on the server. Excess jobs are placed in a job queue. (For further details, see [Job Settings](#)). You can view all jobs that are currently

queued on your server and cancel or start a queued job without having to restart the server or interrupt other jobs.

To view queued jobs:

- Go to **Status > Queued Jobs**. All jobs currently queued on the server are listed.

Details about each job include:

| Details: | Description: |
|----------------|--------------------------------------------------------------------------------------------------|
| Queue Position | Current priority position for each job in the queue. Jobs with a lower queue position run first. |
| Protocol | Name of protocol associated with job. |
| User | Name of client user who launched job. |
| Client | Name of client machine where job was launched. |
| Start Date | Date when job was placed in the queue. |
| Update | Buttons allow you to start or end a queued job. |

To start a queued job:

- Click **Run**  for the listed job.

To cancel a job:

- Click **End Job**  for the listed job.

Tip: To adjust the rate at which the list is updated, enter a different value in **Refresh Rate** and click **Set**. Default is 15 seconds.

Running Jobs

When a client runs a protocol, the running protocol and its resulting data are known as a "job". It is common to run multiple jobs from the same protocol. You can view all jobs that are currently running and cancel a job without having to restart the server or interrupt other jobs.

To view running jobs:

- Go to **Status > Running Jobs**. All jobs currently running on the server are listed.

Details about running jobs includes:

| Details: | Description: |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------|
| PID | Process Identifier (same as PID in Windows Task Manager). A numerical identifier that uniquely distinguishes a process while it runs. |
| Protocol | Name of protocol associated with job. |
| User | Name of client user who launched job. |
| Client | Name of client machine where job was launched. |
| %CPU | Percentage of the total CPU being used by this job. |
| Memory (Kb) | Number of Kb memory currently being used by this job. |

| Details: | Description: |
|------------|---------------------------------|
| Start Date | Date when job started running. |
| End Job | Option to cancel a running job. |

To cancel a job:

- Click **End Job**  for the listed job.

Tips:

- The running job list is updated at regular intervals. The amount of time between updates is determined by the **Refresh Rate** (default setting is 15 seconds).
- If clients submit jobs that the server cannot process, they are listed above the running jobs. This list is intended to help you identify problems with specific protocols or web clients.

Job Settings

You can customize how protocol jobs run on your servers. Default settings are applied during installation and you can modify them to better support your hardware and software configurations. The job performance settings that you can change are described below.

To change a job performance setting:

1. Go to **Setup > Job Settings**.
2. For each setting you need to modify, change the value.
3. Click **Save**.

| Setting | Description |
|----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Running Job Limit | To prevent server overload, this option limits the number of jobs running on the server at the same time. The default maximum number of jobs that can run simultaneously is 10. Any excess jobs are placed in a queue and processed in order of submission, as running slots become available. To disable this job queuing feature, set the value to "0". |
| Block Job Timeout | Length of time (in seconds) before blocking jobs time out and return an error. The default is 10 seconds. |
| Job Priority Switching | When enabled, job processes are downgraded to a lower priority after 10 seconds. This feature is enabled by default. Normally, this setting does not significantly impact performance. |
| Intermediate Web Port Jobs Release Delay (seconds) | The time to delay the removal of intermediate Web Port jobs after they are run. All jobs that generate forms displayed in the left Web Port pane as well as jobs launched from the right Web Port pane are considered "intermediate". The value suffix can be "d" for days, "h" for hours or "m" for minutes. Set this to -1 or 0 to delay the removal of intermediate Web Port jobs forever. Note: The Pipeline Pilot administrator will need to clean up permanently delayed jobs. |

| Setting | Description |
|---------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Maximum Archived Jobs per Pipeline Pilot User | The server will actively delete jobs that exceed this maximum. Jobs that have an explicit or implicit expiration are excluded from the count. This includes blocking jobs and jobs that have been set in the client to expire. |
| Maximum Number of Simultaneous Parallel Processing Subprotocol Jobs | Maximum number of simultaneous jobs that can be spawned to support each individual parallel processing subprotocol. When the server hosts a parallel subprotocol operation, this setting is used by the running protocol to handle x-number of subprotocol jobs for a specific subprotocol. The default number of jobs is four (4). Increase the value based on the size of your machine and the number of jobs you need to run. |
| Maximum Number of Persistent Job Daemons per Job Pool | Maximum number of persistent daemons associated with a particular job pool. A single job or a set of related jobs can use daemons from the same job pool. After a pooled job finishes running, a daemon remains resident in memory for a configurable period of time waiting for new job requests. Running a job in a daemon is much faster because it skips initialization of many internal data structures. When the server is using a high amount of memory, no new daemons are created. The default maximum number of persistent daemons per job pool is 16. No daemons are created when the setting is set to 0. |
| Persistent Daemon Timeout | Maximum time (in seconds) that an idle daemon will remain resident in memory before it is shut down. The default timeout is 300 seconds (or 5 minutes). |
| Job Readiness Refresh Rate | To keep the server warm during periods of inactivity, a single protocol runs periodically to keep system files in memory and improve performance when starting new jobs. This manifests as an additional scisvr process. The configuration time, in seconds, sets the amount of time in between each protocol execution. The default is 300 seconds (or 5 minutes). To disable this feature, set the value to 0. |
| Pre-started Daemons for Non-Pooled Jobs | (Windows only) When set to a number > 0, a pre-initialized daemon is used to run jobs that are not assigned to any job pool. |
| Use Braces in Job Directory Names (Compatibility) | Generates braces for job directory names (to support backwards compatibility with previous server versions). If you have grids or clusters deployed on a Linux environment, the use of braces in paths may cause problems with third-party software. If your configuration uses grid engines, we recommend setting this option to "No". |
| Maximum Memory Usage of an Individual Job Process | Maximum amount of memory allocated on the server for processing jobs. All scisvr executables that exceed this limit are killed. Valid values include a percentage of total physical server RAM (for example, 75%), and raw number of gigabytes (for example, 16). |
| Maximum Job Age Based on Job Folder Size | Clean up users' jobs based on job age and minimum folder size in KB, MB, and GB. You can add multiple rows to define the maximum age for different folder sizes. Jobs that are set to expire at a future date are ignored. Cleanup is scheduled every four hours. |

Tips:

- You can specify where on the file system the job directories are created (go to **Settings > Folder Locations > Jobs Directory**).
- On clusters and grids, you can improve performance by allowing a job on a local disk to store temporary files (go to **Settings > Folder Locations > Local Temp Directory**).

Note: When using this option all temporary files will be removed when the job completes.

- Some settings may not apply due to your hardware and software configurations.
- For more details on customizing job settings to optimize your enterprise deployment, see "Server Performance Tuning" in the *Pipeline Pilot System Requirements Guide*.
- By default, the server keeps up to 100 jobs for each client user (Pipeline Pilot Client). Older jobs are deleted when the client reaches its maximum allotment. To increase or decrease the maximum number of jobs maintained on the server, go to [Setup > Server Configuration](#), and change the setting for **Maximum Archived Jobs per Pipeline Pilot User**.
- Web Port does not remove client jobs. The Web Port help recommends that users remove older files no longer needed from the Jobs tab.

Job Folder Maintenance

A "job" is an instance of a running process, plus an organization of folders and files that contain all the data related to that protocol's execution. Once the running process completes, the job folder may remain, and the lifetime of the job folder is determined by which interface was used to launch the job initially.

Job Folder Lifetimes

Here are details on various job folder lifetimes for jobs created from various Pipeline Pilot interfaces:

- **Pipeline Pilot Client:** Pipeline Pilot jobs are created with a jobdir whose name starts with the letters "ppc", for which a set number are kept, depending on a combination of personal settings for number of jobs to save in the client (**Tools > Options > Jobs tab list size**) and the **Maximum Archived Jobs per Pipeline Pilot User** setting in the Admin Portal (**Setup > Job Settings**). Job folders outside these limits are deleted when the client session is closed.

Note: The Admin Portal limit applies only to jobs created in Pipeline Pilot Client.

- **Web Port:** Web Port jobs are created with a jobdir whose name starts "wpt". Job folders are kept until the user explicitly deletes them from the Web Port Jobs tab, allowing users the ability to re-run previously launched jobs. Web Port users should get in the habit of occasionally looking at the Jobs tab and deleting protocols they no longer wish to reuse. (Web Port may also create temporary utility job folders without this job folder naming scheme).
- **SharePoint Bridge:** For jobs invoked via the SharePoint Bridge client, a session cookie is typically stored and used. Session cookies default to expiring after 30 days, and can be customized in the Admin Portal setting "Expiration (days, hours or minutes) of Single Sign-on Credentials" (**Setup > Server Configuration**). After the session cookie expires, previously run jobs may not be available for viewing from SharePoint. The session cookie is renewed each time a protocol on a SharePoint page is explicitly re-run.
- **Blocking Web Services:** By default blocking jobs are deleted when the job completes. Strictly speaking, they are marked for deletion with an expiry that is either zero, or 10 minutes if the web service includes at least one job folder file in its result. This is so that the web service client has some

time to retrieve the job results before it is deleted. Note that there is a distinction between what files a protocol may write out and what a web service wrapper may define as the result. It is the web service definition that is used to judge whether a job can be removed immediately (because its data results have been returned in the service response) or needs to be retained for a short period.

The default expiration behavior can be overridden by web service clients or by web services themselves by extending the expiry to any amount or by having no automatic expiry. In this latter case, the behavior is more like that of a non-blocking service (see below).

Job expiry is mediated by a file named ".expiration" in the "RunInfo" subfolder of the job. The 3rd line of the file indicates in human readable form when the job is set to expire (0 means "Now").

Completed jobs that are expired (i.e., jobs that have expiration files whose date has passed) are eligible for cleanup. The clean up sweep happens very frequently, so expired jobs will not exist for more than a couple of seconds after expiry.

- **Non-Blocking Web Services:** In the case of a non-blocking (or asynchronous) service, the job is launched by the server and then left to run. The client will typically monitor its progress and retrieve results when complete. The client can then delete the job using an appropriate API, depending on the SDK or API they are to run and manage job. Non-blocking jobs can also be run with an expiration behavior, instead of being left for the client to delete.
- **Runprotocol.exe:** Job folders are immediately deleted unless the -GUI flag is used. When the -GUI flag is used, at job process completion, the job folder is deleted after the specified seconds have elapsed. If a value is not specified, you are prompted for a key press to delete the job.
- **SDKs:** Job folders created by Pipeline Pilot SDKs (for example, Java and .NET) are created with no expiry. The Pipeline Pilot developer is expected to call functions such as `pp.ReleaseJob` explicitly to handle the cleanup.
- **Discovery Studio:** For protocols that run via Discovery Studio, the jobs are deleted from the server once the user selects them for download to Discovery Studio.

Because these job folders, particularly for jobs invoked via Web Port, can accumulate over time and take significant storage space, Pipeline Pilot provides the ability to redirect the Jobs folder to a larger drive location, configured in the Admin Portal settings (**Setup > Folder Locations**).

If this is not possible or insufficient, you might institute a policy where job folders older than a set age are removed, and users are warned that this will occur. Deleting older jobs folders prevents users from being able to see the previous job settings, protocols and results. It also prevents those jobs from being readily reopened and re-run from (for example, from the Web Port Jobs tab). This does not impact any protocols or data that the user properly stored in the XMLDB or in their user directories. This type of cleanup could be done via a Pipeline Pilot protocol that runs via a scheduled task or cron to delete defined jobs folders older than a certain date.

Maximum Job Age Based on Job Folder Size

In **Setup > Job Settings**, you can set a **Maximum Job Age Based on Job Folder Size**. This setting controls clean up of users' jobs based on job age and minimum folder size in KB, MB, and GB. You can add multiple rows to define the maximum age for different folder sizes. Jobs that are set to expire at a future date are ignored. Cleanup is scheduled every four hours.

Scheduled Tasks

Overview

A scheduled task can be any of the following events:

- User-defined protocol jobs.
- Administrator-defined tasks that need to run on a regular basis, such as XMLDB backups.
- Package-defined administration tasks, such as re-indexing a data source.
- Predefined platform housekeeping tasks that run on the server (that are not parameterizable). These types of tasks typically involve events such as system cleanup or cache refresh operations (including the refresh operation for the scheduled task cache).

Scheduling Tasks

In Pipeline Pilot a trigger-based scheduler allows you to schedule various types of tasks that are triggered at predefined times and frequencies, running on the server with predefined parameters. See the *Application Packaging Guide* in the Developers Help for details about setting up scheduled tasks.

Monitoring scheduled tasks

You can monitor these scheduled tasks to find out their last and next scheduled run times since the server was started.

To monitor scheduled tasks:

1. Log into the Admin Portal.
2. Go to **Status > Scheduled Tasks**. All scheduled tasks currently defined on the server are displayed in a table.

The following information is displayed for each task:

- **Task Name:** The name assigned to each task. This is the name of the task definition file with the file extension removed.
- **Owner:** Identifies the task owner. Platform indicates platform tasks. Package-defined administration tasks start with "scitegic/", followed by the package name (for example, "scitegic/appcatalog").
- **Type:** Task types include Protocol and Service.
- **Specification:** Provides the execution details about the task. For a Protocol type task, the protocol name is displayed. For a Service type task, the service name and related action is displayed.
- **Schedule:** Indicates how the scheduled task is triggered on the server:
 - **Interval:** Tasks that occur repeatedly at regular time intervals, (for example, 1 h, 10 m, 2 s).
 - **Cron:** Tasks that occur repeatedly at the times that are predefined in a cron specification, (for example, at 1:00 AM every weekday).
- **Last Run:** Date and time the scheduled task was last run since the server's latest start.
- **Next Run:** Date and time for the next scheduled task.

To sort the list of tasks:

1. Point at the right side of a column header to display the dropdown.
2. Select **Sort Ascending** or **Sort Descending**.

To hide or show columns:

1. From the column dropdown, select **Columns**.
2. From the submenu, check or uncheck the column names.

Displaying Server Information

Server Information is a support tool that you can use to quickly locate information about your Pipeline Pilot server including:

- Environment
- Ports
- Paths
- Tomcat System Properties

Environment

Use Environment to:

- Get details about the operating system where your Pipeline Pilot Server is installed.
- Look up the name of the user logged into the server.
- Check the process IDs for your Apache server

Note: The *Login Name* is the name under which the Apache server is running. If network access is required, do not set this to the local system user on Windows.

Ports

Use Ports to check port numbers for the following servers:

- SSL
- Apache server (HTTP ports)
- Tomcat server (shutdown and HTTP ports)
- Derby server

Tip: For more information on changing the ports configuring during installation, see [Reconfiguring Ports](#).

Paths

Use Paths to:

- Determine where the Apache server software is installed
- Get detailed information and statistics about the Apache server
- Determine where the Tomcat server software is installed

Tip: For more information on managing your Java server, see [Configuring Java Servers](#).

Monitoring Server Usage

Use the Server Status page (Status > Server Status) to monitor the status of your Pipeline Pilot Server. It provides some general metrics related to your server, including details about load levels, CPU and memory usage.

This information should help you identify the load on the system and the availability of the server to run further protocol jobs. The load is expressed in terms of CPU usage (current and recent measurements) and memory usage (physical and swap). Server load is defined according to the following levels:

| Server Load Level | CPU Usage | Physical Memory Used | Total Memory Used |
|-------------------|-----------|----------------------|---------------------|
| Critical | > 90% | > 85% | > 75%5 |
| Poor | > 75% | > 70% | > 65% |
| Medium | > 50% | > 60% | |
| Good | > 20% | Within above limits | Within above limits |
| Excellent | < 20% | Within above limits | Within above limits |

Tips:

- You can also view resource usage related to protocol job processes running on the server.
- For a Linux server cluster, the page includes a list of secondary nodes. You can refresh the server status information for individual servers or for all servers using the update buttons in the secondary node table.

Chapter 3:

Reports

Monitoring XMLDBs with Catalog Search

Overview

Use Catalog Search to search the XMLDB of one or more servers for protocols and components. For example, when preparing for an upgrade, you can identify most/least frequently used protocols and find protocols that are authored by specific users.

As a general server administration tool, you can query for references about specific servers, authors, and protocol parameters. Catalog Search supports constructing ad hoc queries that result in a set of protocol names with associated protocol and user information.

Notes:

- The search index is updated on a regular basis. You can make changes to the scope and frequency of the indexing from the [Catalog Settings](#) page.
- If you change the port settings for the server, you will need to re-index the catalog. See [Catalog Settings](#).

Using Catalog Search

To search the catalog:

1. Go to **Reports > Catalog Search** and specify the text to search. Leave the **Remote Servers** field blank to search only the current server, or specify additional comma-separated servers on your network to search.

Note: The Remote Server search will only work for servers that are registered. See [Server Registry](#).

2.

QueryResults

Text search: red

Remote Servers: vcl-rh5-1-sd:9604/, pps-prod-win:9603

Protocol Queries

Author: Select or add author here, separated by commas

Date last changed: From: To:

Type: ☒ All ☐ Protocols ☐ Components

Advanced Protocol Queries

Usage Queries

Number of runs: Min: Max:

Last run date: From: Until:

Run by: Select or add user here, separated by commas

Reset Query

Search

Page 32 | Pipeline Pilot 2021 • Admin Portal Guide

3. Click **Search**.

| Query Results | | | | | | |
|------------------------------|------------------------|-----------------|-------------------|-------------|----------|----------|
| Export to Excel File | | | | | | |
| Protocol | Path | Package | Last Saved Date | Server | Author | Last Run |
| Intensity Reduction of ... | Protocols/Examples/... | scitegic/ima... | 2012-08-15 18:... | bioregpp... | Accelrys | |
| Unpack Color | Components/Imagin... | scitegic/ima... | 2013-01-03 14:... | bioregpp... | Accelrys | |
| Using Find Peaks | Protocols/Examples/... | scitegic/ima... | 2012-11-08 18:... | bioregpp... | Accelrys | |
| Using Shrink Objects | Protocols/Examples/... | scitegic/ima... | 2012-10-03 14:... | bioregpp... | Accelrys | |
| Using Skeletonization | Protocols/Examples/... | scitegic/ima... | 2012-11-08 17:... | bioregpp... | Accelrys | |
| Creating Mask for Regi... | Protocols/Examples/... | scitegic/ima... | 2012-12-19 11:... | bioregpp... | Accelrys | |
| Repacking Color | Protocols/Examples/... | scitegic/ima... | 2012-08-15 19:... | bioregpp... | Accelrys | |
| Classify for Classificati... | Protocols/Examples/... | scitegic/ima... | 2011-05-19 17:... | bioregpp... | Accelrys | |
| Using Watershed from... | Protocols/Examples/... | scitegic/ima... | 2012-12-19 12:... | bioregpp... | Accelrys | |
| Using Hit Miss Transform | Protocols/Examples/... | scitegic/ima... | 2012-11-08 17:... | bioregpp... | Accelrys | |
| Using Morphological R... | Protocols/Examples/... | scitegic/ima... | 2012-11-08 17:... | bioregpp... | Accelrys | |
| Using Flood Fill | Protocols/Examples/... | scitegic/ima... | 2012-11-08 17:... | bioregpp... | Accelrys | |
| Using Find Template | Protocols/Examples/... | scitegic/ima... | 2012-11-08 18:... | bioregpp... | Accelrys | |
| Segment Spots on Circ... | Protocols/Examples/... | scitegic/ima... | 2012-11-08 17:... | bioregpp... | Accelrys | |
| Two Image Math Exam... | Protocols/Examples/... | scitegic/ima... | 2012-08-20 14:... | bioregpp... | Accelrys | |
| Using Draw Shapes | Protocols/Examples/... | scitegic/ima... | 2012-08-07 16:... | bioregpp... | Accelrys | |
| Watershed Segmentat... | Protocols/Examples/... | scitegic/ima... | 2012-12-21 13:... | bioregpp... | Accelrys | |
| Using Find Straight Lines | Protocols/Examples/... | scitegic/ima... | 2013-03-05 17:... | bioregpp... | Accelrys | |
| Using Level Sets | Protocols/Examples/... | scitegic/ima... | 2013-03-02 11:... | bioregpp... | Accelrys | |
| Using Image Toggle Vi... | Protocols/Examples/... | scitegic/ima... | 2013-03-03 23:... | bioregpp... | Accelrys | |

4. Click the **Query** tab to return to the original search options or click **Export to Excel File** to save the results.

Query Options

You also can refine a search by using the Protocol Queries, Advanced Protocol Queries, and/or Usage Queries fields.

Protocol Query Options

- **Server:** Search for files published on specific servers.
- **Author:** Search for files authored by specific users.
- **Date last changed:** Search for files that were modified within a specific date range.
- **Type:** Filter the type of files to include in the results (Protocols, Components, both).

Advanced Protocol Query Options

- **Number of versions:** Every time a file is saved in the XMLDB, a version of the file is saved based on the latest changes. Some files might have numerous versions while others have only a few. You can locate files based on how frequently (or infrequently) they are updated by specifying minimum and

maximum number of versions.

- **Parameter:** Locate only files that contain specific parameters and parameter values.

Note: The search engine has a list of special characters: + - && || ! () { } [] ^ " ~ * ? : \. When searching parameter values that equal or contain any of these special characters, the search may not result in any hits. To work around this issue, change the query to the parameter that contains part of the value without any of the above mentioned special characters.

Usage Query Options

- **Number of runs:** Filter protocols based on how many times jobs were processed.
- **Last run date:** Filter protocols based on date ranges for when the jobs were processed.
- **Run by:** Filter protocols based on user-submitted protocol jobs.

Exporting Results to Excel

From the search results, click **Export to Excel File**  .

Tips:

- This file can be further filtered in Excel and shared with appropriate end users.
- You also can use the information as input to an analysis protocol (e.g., create and post custom reports or send email to authors of specific protocols).

Completed Jobs

The Admin Portal tracks all jobs submitted by clients that run on your servers. A log file is available that provides details about each submitted job. This information is useful for tracking the history of submitted jobs and to identify problems.

To view the job log:

1. Go to **Reports > Completed Jobs**. A list of all jobs submitted to the server is displayed.
2. By default, jobs are sorted by start time (the most current listed first). To change the sort order, click any column header. You can also use filters to further sort job listings.

The following details are provided for each submitted job:

| Job Info: | Details: |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Job | Name of protocol associated with the submitted job. A protocol can run multiple times. For each instance, a separate job is issued on the server |
| Job Version | Protocol version associated with the job. Each time a protocol is saved, it is saved as a "version" in the XMLDB |
| User | Name of user running the job |
| Client | Name of client machine that submitted the job |
| Server | Name of server processing the job |
| CPUs Used | CPUs Available Number of CPUs available when the job was run (invariant across jobs) |

| Job Info: | Details: |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Start Time | Date when job started running |
| Duration [sec] | Amount of time that elapsed while job was running (in seconds) |
| Status | <p>Status indicator for job, including:</p> <ul style="list-style-type: none"> ■ Finished: Job was successfully completed. It ran without errors and generated results. ■ Stopped: User stopped the running protocol before it was completed. Job did not generate results. ■ Canceled: An administrator canceled the running protocol before it was completed. ■ Error: Job could not be executed or processed due to a problem within the protocol, (e.g., an error with a parameter value or script) or due to a data processing error on the server, (e.g., data wasn't accessible at the time the job was submitted). Error messages are displayed on the client to help the user isolate the source of the problem and debug the component if necessary. |

Filtering the Jobs Log

You can filter the jobs log to narrow down the details. This is useful if you want to view or print a subset of records for tracking a range of jobs. For example, you can view jobs for specific users or jobs within a certain name.

To filter jobs:

1. In **Filter by**, select a filter option (e.g., User).
2. Specify what kind of action you want to perform on the filter option (e.g., view jobs for a specific user name).
3. Click **Filter**. The list of jobs is updated to display only the ones that match your filter criteria.
4. You can filter on filtered records by selecting more options. Each set of filters you enter is displayed on a separate line above the records, so you can track what information is shown.

Tips:

- To revert to the default job display (by most current date), click **Reset**.
- By default, 100 jobs are displayed. To view more or fewer jobs, change the **Job List Size**. (It takes more time to display many jobs.)

Installed Collections

Pipeline Pilot functionality is licensed as predefined groupings of related packages, known as collections. Collections include constituent packages. (A package is a method for organizing products to facilitate collection installation, removal, and upgrades.)

You can look up details about any installed collection on your server.



To view details about installed collections:

1. Go to **Reports > Installed Collections**.

The collections installed on your server are identified in a list (organized by collection name and version).

| Collection Name | Version |
|--------------------------------------------|---------|
| + Platform | 9.0.1 |
| + Chemistry Collection | 9.0.1 |
| + ADMET Collection | 9.0.1 |
| + Data Modeling Collection | 9.0.1 |
| + Imaging Collection | 9.0.1 |
| + Sequence Analysis Collection | 9.0.1 |
| + Gene Expression and Mass Spec Collection | 9.0.1 |
| + Documents and Text Collection | 9.0.1 |
| + Lab Analytics Collection | 9.0.1 |
| + Polymer Property Prediction Collection | 9.0.0 |
| + Materials Studio Collection | 9.0.1 |
| + Pipeline Pilot | 9.0.1 |

2. To find constituent packages for an installed collection, click  to expand the listing.

| Collection Name | | Version |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
|  | Platform | 9.0.1 |
| | Accelrys/Admin Portal 9.0.1 Accelrys/Admin Portal Container 9.0.1 Accelrys/Accelrys Catalog 9.0.1 Accelrys/Client SOAP SDK 9.0.1 Accelrys/Client SDK - .NET 9.0.1 Accelrys/Client SDK - Java 9.0.1 Accelrys/Client SDK - JavaScript 9.0.1 Accelrys/Core 9.0.1.22 Accelrys/Core Utilities 9.0.1 Accelrys/Data Access 9.0.1 Accelrys/Platform Documentation 9.0.1 Accelrys/Data Source Builder 9.0.1 Accelrys/GEMS 9.0.1 Accelrys/GEMS Application 9.0.1 Accelrys/Generic 9.0.1 Accelrys/Integration 9.0.1 Accelrys/Java Server 9.0.1 Accelrys/App Launcher 9.0.1 Accelrys/Mongo DB 9.0.1 Accelrys/Query Service 9.0.0 Accelrys/Reporting 9.0.0 Accelrys/Reporting (Web Apps) 9.0.1 Accelrys/RSS Integration 9.0.1 Accelrys/Scheduler 9.0.1 Accelrys/SharePoint Bridge 9.0.1 Accelrys/Web Application Framework 9.0.1 Accelrys/Web Port 9.0.1 Accelrys/Web Services 9.0.1 | |
|  | Chemistry Collection | 9.0.1 |

Tips:

- If a collection is lacking one or more of its required packages, it is indicated by an italic font (e.g., *Platform*).
- Packages are either absent or embedded.
- An absent package is indicated like this: ~~BIOVIA/ADMET~~.
- An embedded package is indicated like this: Accelrys/Web Application Framework 9.0.0.
- An embedded package is a type of data, Java class, global property or other set of files that resides in an unlicensed package on your server. This information is required by some of your licensed packages and the Admin Portal tracks these dependencies so you know what other files are required on your server. Even though an embedded package is installed, there is no Pipeline Pilot Server access to its component set.

Installed Packages

All installed component collections are organized on the server by "package". The package provides instructions for installation, removal, and upgrades. A package may include components, protocol examples, example data, Perl modules, Java classes, VBScript libraries, binary executable files, scripts, APIs, and documentation. It can also include custom scripts and commands invoked during publication or removal.

You can look up details about any installed package on your server.

To view details about installed packages:

- Go to **Reports > Installed Packages**.

The package information includes:

- Vendor
- Package name
- Version number
- Active Status
- Installation date
- Expiration date
- Editable

| Vendor | Name | Version | Active? | Installation Date | Expiration Date | Editable |
|-------------|-------------------------------------------|-----------|---------|--------------------------|-----------------|----------|
| SciTegicDev | Admin regression tests conf | 17.1.0.90 | Yes | Thu May 05 11:52:41 2016 | 30-Apr-2020 | |
| SciTegicDev | Admin command line config regression test | 17.1.0.90 | Yes | Thu May 05 11:52:48 2016 | 30-Apr-2020 | |
| SciTegicDev | Admin Portal Dev | 17.1.0.90 | Yes | Thu May 05 11:52:49 2016 | 30-Apr-2020 | |
| SciTegicDev | Admin regression tests security conf | 17.1.0.90 | Yes | Thu May 05 11:52:51 2016 | 30-Apr-2020 | |
| SciTegicDev | App Catalog Regression Testing | 17.1.0.90 | Yes | Thu May 05 11:52:53 2016 | 30-Apr-2020 | |
| SciTegicDev | Common Entity Management (CEM) | 17.1.0.90 | Yes | Thu May 05 11:52:59 2016 | 30-Apr-2020 | |

Notes:

- Editing rights for installed packages are managed in the [Package Editors](#) page. If users or groups are granted rights to edit a package, a **Details** button appears in the Editable column that links to the **Package Editors** page.
- A status of "Yes" means the package is active on the server. "No" means the package is either inactive or embedded.
- An "embedded package" is a type of data, Java class, global property or other set of files that resides in an unlicensed package on your server. This information is required by some of your licensed packages and the Admin Portal tracks these dependencies so you know what other files are required on your server.

Server Usage Report

The Server Usage Report provides information to help you monitor how your Pipeline Pilot Servers are used within your organization. It provides statistics about client usage, collection usage, protocol consumption, and job submissions.

There are two sets of options:

- **Servers:** Use these options to specify which servers to monitor in the report. (Available servers are based on what is specified in the [Server Registry](#).) The server list always contains at least one name, since the server you logged into is identified as "Local Server".
- **Report Options:** Use these options to specify the range of dates the report should cover and how to display user names and license information.

Notes:

- When you run a report that includes a specific server, the log file from that server is copied into a local log file cache. On subsequent occasions, the file is only copied if it is newer than the cached copy.
- In some circumstances, you may need to copy a log file manually into the local cache, (e.g., when a server is no longer running, when a log file is transferred from a remote site). The filename should match the server name from which it originates, since this name is used to identify the log file in the list.

To build a report:

1. Go to **Reports > Server Usage Report**.
2. In **Servers**, specify each server you want to include in the report. (Checked servers are included in the report. Unchecked servers are excluded.)
3. In **Report Options**, specify the time frame for the report by entering a **Start Date** and an **End Date**.
4. To hide the identities of users in the output, check **Anonymize user names**. The default setting is to reveal user names in the report.
5. Click **Build Report**.

Report Content

Your report content includes:

- Date range for which the report data is published
- Usage statistics listed by component collection and/or application

- More detailed usage statistics organized by user and type of client (for web clients)
- Server names and Pipeline Pilot Server version numbers included in the reported usage statistics

Usage Categories

Usage categories are based on the following:

- **Frequent user:** Runs jobs on the server at least 30 days per year
- **Infrequent user:** Runs jobs between 3 and 29 days per year

Note: For brief reporting periods, these categorizations are not meaningful.

Validation Report

Overview

Pipeline Pilot uses a set of validation rules that include security checks, best practice guidelines on protocol and component design, and documentation and style guidelines. You can customize the validation rules to incorporate standards that are enforced or promoted within your organization. (For details, see [Configuring Validation Protocols](#).)

Administrators should validate XMLDBs on a regular basis to ensure that deployed protocols and components meet certain security requirements and follow design standards. Use the Protocol Database Validation feature for this purpose. It applies a set of validation rules to all published, non-packaged protocols and components in the XMLDB and generates a summary of validation results in a report.

Customizing the Validation Schedule

XMLDB validation is configured to run as a scheduled task by default. You can customize the schedule to suit your preferences.

To customize the validation schedule:

1. Go to **Reports > Validation Report**.
2. To set the frequency (days and times) for performing the validation:
 - Check the specific days of the week.
 - Enter values for hours and minutes. Use a 24-hour clock, (e.g., enter "14" for 2:00 PM). To update more than once per day, enter multiple times separated by commas. For minutes, enter minutes past the hour (1-60).
3. Click **Save Schedule**.

Schedule XMLDB validation job

Days: ☒ Sun ☒ Mon ☒ Tue ☒ Wed ☒ Thu ☒ Fri ☒ Sat

Hours: Minutes past hour:

To disable a scheduled validation task:

- Click **Disable Job**. (An Enable job button is displayed whenever you disable a job, in case you decide to resume the validation as a scheduled task.)

To resume a scheduled validation task:

- Click **Enable Job**.

Save Schedule

Enable Job

Manually Running a Validation

To manually run a validation:

- Click **Revalidate** or **Refresh** (upper-right corner of Validation Summary table).

Tip: If this task was not previously done on your server, click **Validate Now**. It might take several minutes for the validation to complete.

Validation Report Results

The Validation Report output includes a summary of all violated validation rules and a count of occurrences in components and protocols for each. Results are organized into the following columns:

- Validation rule
- Severity
- Number of protocols

| Validation Summary - Report generated on 02/20/2013 11:58:35 | | | Revalidate |
|-----------------------------------------------------------------------------|----------|---------------------|----------------------|
| Validation Rule | Severity | Number of Protocols | |
| Design: Shortcut has no reference component | Error | 1 | |
| Documentation: Undocumented parameter legal values | Warning | 1 | |
| Style: Help text contains "this component" | Info | 1 | |
| Style: Summary help text appears not to start with 3rd person singular verb | Info | 1 | |
| Style: Summary help text identical to component/protocol name | Info | 1 | |
| | | | Export to Excel File |

You can sort the columns in ascending and descending order, show/hide the columns to display, and export the data into a spreadsheet.

| Validation Summary - Report generated on 02/20/2013 11:58:35 | | | Revalidate |
|-----------------------------------------------------------------------------|----------|---------------------|------------|
| Validation Rule | Severity | Number of Protocols | |
| Design: Shortcut has no reference component | | 1 | |
| Documentation: Undocumented parameter legal values | | 1 | |
| Style: Help text contains "this component" | | 1 | |
| Style: Summary help text appears not to start with 3rd person singular verb | | | |
| Style: Summary help text identical to component/protocol name | | | |

Sort Ascending
Sort Descending

Columns
Info

☒ Validation Rule
☒ Severity
☒ Number of Protocols

Exporting Results to Excel

To obtain a full report that describes all validation errors in more detail:

- Click **Export to Excel File** (lower-right corner of Validation Summary table).

Export to Excel File

ValidationRepo

| | A | B | C | D | E | F | G | H | I |
|---|----------|--------------|------------|--------------|-------------|----------|-------------|--------------------------|---|
| 1 | Path | Name | Author | LastModified | Validation | Severity | Details | ComponentDisplayName | |
| 2 | Componer | HitList to C | shawn.lavi | Wed Feb 1 | Style: Sum | Info | The summ | HitList to Query | |
| 3 | Componer | HitList to C | shawn.lavi | Wed Feb 1 | Style: Sum | Info | The summ | HitList to Query | |
| 4 | Componer | HitList to C | shawn.lavi | Wed Feb 1 | Style: Help | Info | Help text c | HitList to Query | |
| 5 | Componer | HitList to C | shawn.lavi | Wed Feb 1 | Document | Warning | The help f | HitList to Query | |
| 6 | Componer | HitList to C | shawn.lavi | Wed Feb 1 | Document | Warning | The help f | HitList to Query | |
| 7 | Componer | HitList to C | shawn.lavi | Wed Feb 1 | Design: Sh | Error | The refere | Execute Query Definition | |

Tip: This file can be further filtered in Excel and shared with colleagues.

Foundation Applications

You can use the **Foundation Applications** report in Pipeline Pilot to view the list of registered Foundation applications and update it manually if needed. Some BIOVIA applications are detected and registered automatically. However, if Pipeline Pilot packages are modified as part of an application installation and you will need to register them manually. To do this click **Update Applications** at the top-right corner of the table.

Chapter 4:

Security

Security Overview

Pipeline Pilot includes enhanced security capabilities to support the following:

- Authentication
- Impersonation
- Authorization

Authentication

The following authentication features are supported on the platform:

- Authentication against a corporate user directory, such as Active Directory.
- Authentication against a list of users defined by the administrator. This can be used alone or in addition to authentication against a corporate user directory.
- Kerberos/SPNEGO enhancements
- Some SAML support for SOAP-based web services
- 3D Passport
- JAAS

Pipeline Pilot offers several ways to authenticate, based on your security policies and how you want to control access. Supported authentication methods include:

- **User List:** A list of users and passwords that can be used to log into the server. (Can be enabled with other methods.)
- **External directory of users:** Authenticate against a set of user credentials not maintained by Pipeline Pilot, (for example, corporate directory). This feature can be used in conjunction with a Pipeline Pilot User List and supports Pipeline Pilot options such as Any User Name, Local, and Domain.

Note: With all available authentication methods, the server creates a session for the user that includes the permissions defined for that user.

Support for SAML Authentication

Pipeline Pilot includes some support for Security Assertion Markup Language (SAML) standard, when the client server communication is via SOAP. The WS-Security standard defines how security information can be included in a SOAP header; when this takes the form of simple user names and passwords (Username Token profile), the credentials are validated against the authentication method in force for the Pipeline Pilot Server.

In addition, the Pipeline Pilot SOAP handler service supports the passing of SAML-based authentication data (Binary Security Token profile) in the SOAP header. To enable this, you must import SAML certificates into a Trusted Certificate store for those servers from which Pipeline Pilot should accept SAML assertions in SOAP-based requests.

Impersonation

Impersonation provides running protocol jobs with exclusive user-based access to resources and, conversely, prevents excess permission from being granted to a single utility user to cover all resource

access. With impersonation enabled, a server can run protocols under the client's user name instead of the server account name. Clients can then use their network security credentials, instead of the server account credentials, to access network resources under the precise privileges assigned to their user name.

Pipeline Pilot supports client impersonation where the server runs protocols under a client's user name instead of the server account name.

Both Full and restricted impersonation are supported for running Pipeline Pilot on Windows.

Authorization

In previous releases of the platform software (that is, Pipeline Pilot 8.5 and earlier), security centered around a role-based concept. Since a role is really just a permission to perform a task, (for example, running Web Port), the platform now uses the term "permission" instead of "role". Authorization defines the permissions of client users.

Notes:

- Permissions are assigned to groups of users.
- A group is a collection of users that define a role (for example, platform administrators).
- Package-defined permissions and groups are now supported.

Pipeline Pilot System Permissions

The following default system permissions are available:

- *Platform/Administration/Logon*
- *Platform/PipelinePilot/Logon*
- *Platform/PipelinePilot/Administer*
- *Platform/RunProtocol*
- *Platform/WebPort/Logon*
- *Platform/Logon*

Note: Other BIOVIA-provided packages and collections can also define their own system permissions and groups. Depending on your particular installation of products, additional system permissions might be available.

Pipeline Pilot System Groups

The following default system groups are available:

- *Platform/Everyone*
- *Platform/Users*
- *Platform/PowerUsers*
- *Platform/Administrators*
- *Platform/WebPort/Users*
- *Platform/Denied/Users*

Managing Pipeline Pilot Authorization

Previously for local/domain authentication, it was possible to specify that a user needed to belong to one or more groups for login access on the platform. This was how authorization was handled on the Authentication page in the Pipeline Pilot 8.5 Administration Portal.

Now login access is handled by the *Platform/Logon* permission. To get a logon session and perform tasks on the Pipeline Pilot, the user must have the *Platform/Logon* permission. How it works:

- By default, all users (*Platform/Users* group) have the *Platform/Logon* permission.
- Old behavior is emulated by creating a group, adding members, assigning the *Platform/Logon* permission, and then removing the *Platform/Everyone* group from the *Platform/Users* group. (The `pkgutil` tool automatically handles this for legacy upgrades. Administrators do not need to perform any manual tasks to retain their existing authorization restrictions when performing an upgrade.)
- Groups are added and managed on the **Security > Groups** page. Membership in the group, as well as permissions provided to the group members, can be managed on this page.
- Permissions are added and assigned on the **Security > Permissions** page.

Authorization Guidelines

Follow these guidelines to support authorization on your server:

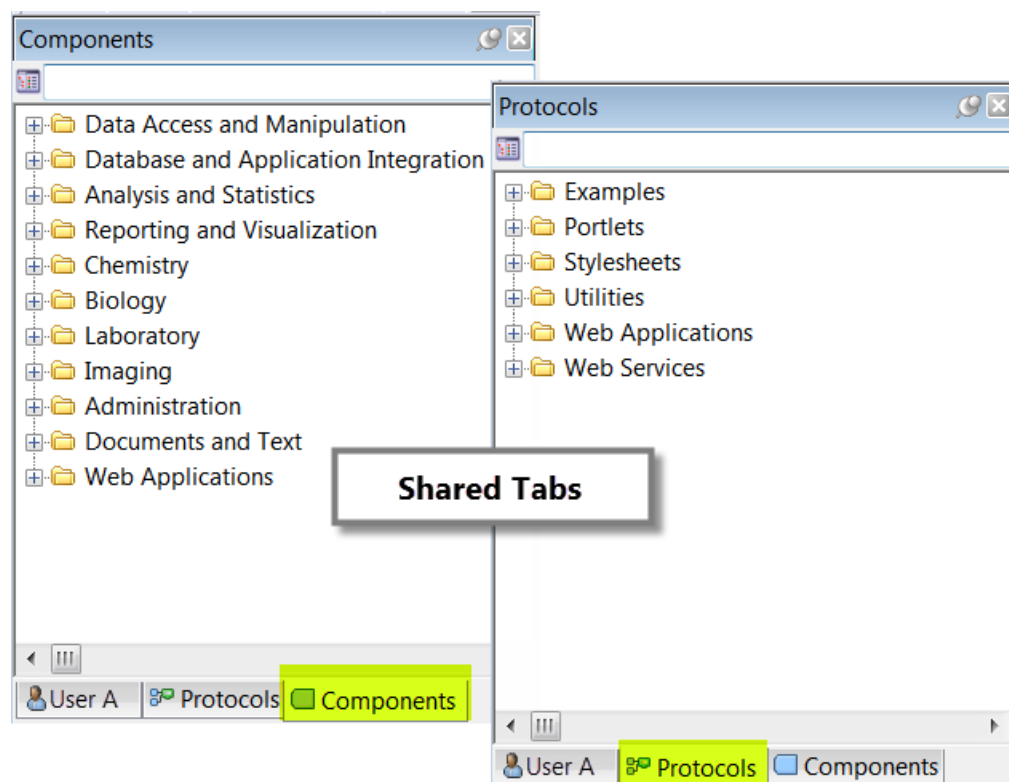
- To ensure that administrators can log into the Admin Portal, always leave *Platform/Logon* in the *Platform/Administrators* group.
- Assign permissions to *groups* (not users).
- Create custom groups that reflect an organization (e.g., *MyCompany Users*).
- Define membership in these groups through operating system groups (typically groups defined in an Active Directory) or by explicit user or group memberships.
- Assign these groups as members to the package-defined groups, (e.g., remove *Platform/Everyone* from *Platform/Users* and replace with *MyCompany Users*).

Access Rights

XMLDB Access Rights Overview

The protocol database (XMLDB) is where all client-accessible protocols and components reside on the server. Whenever protocols or components are read from or written to the server, they are handled in some area of the XMLDB.

The Explorer window in Pipeline Pilot includes two shared tabs (Components and Protocols) and an individual "User Name" tab for each client. Protocols that users create and save for their own personal use are saved to the individual tab. Shared components and protocols are accessible from the shared tabs.



Tab Publishing Rights

By default, all users have read/write access to the shared tabs. When a component or protocol is written to one of the shared tabs in the database (Components or Protocols), it is a "published" protocol. You can control how users publish protocols on these shared tabs by managing access rights. You can control tab access and publishing rights, even specifying where users can read and write in the tab hierarchy, at the folder level.

Managing Access Rights

XMLDB access rights can be configured for both Pipeline Pilot Server and Web Port users.

Administrators can assign exclusive folder write access to some users, leaving the folder contents readable and executable by other users. Some users can have exclusive read permissions, effectively hiding the folder contents from all other users. You can also assign permissions to Pipeline Pilot groups as well as users.

The Access Rights page in the Admin Portal provides a way to manage tab and file permissions (Security > Access Rights). The default setting, which allows users to have *Read/Write* access to the shared tabs, is shown in the following example:

Access Rights Defined

Access rights are defined as follows:

| Permission | Description |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Read/Write</i> | Allows the user or group to read from and write to a folder. If a set of users or groups has <i>Read/Write</i> permission for a folder, everybody else has <i>Read Only</i> permission. |
| <i>Read Only</i> | Allows the user or group to read and execute protocols/components in that folder. If a set of users or groups has <i>Read Only</i> permission for a folder, no one else has access to that folder. |
| <i>None</i> | Folder inherits settings from its parent folder. |

About the *everybody* Group

A special group name is available for all users (named "everybody"). When no specific access rights are defined, *everybody* has *Read* and *Write* access. Administrators can restrict access by assigning explicit access rights to individual users and groups.

Guidelines for Assigning Access Rights

| To: | Do this: |
|----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Assign full access rights to selected entities (users or groups). | Assign <i>Read/Write</i> permissions to all entities that should have full access rights to a folder. Only those entities listed have <i>Read/Write</i> access to the tab. By default, everybody has <i>Read Only</i> access rights. |
| Assign full access rights to selected entities and deny access to everyone else. | Assign <i>Read/Write</i> permissions to all selected entities, and set everybody's permission to <i>None</i> . |

| To: | Do this: |
|--------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Assign Read Only access rights to selected entities (they can read and execute folder contents). | Add the names of all entities that should have rights to read from a specific folder or subfolder. Under this scenario, <i>everybody</i> does not have access to the folder. |

To configure access rights:

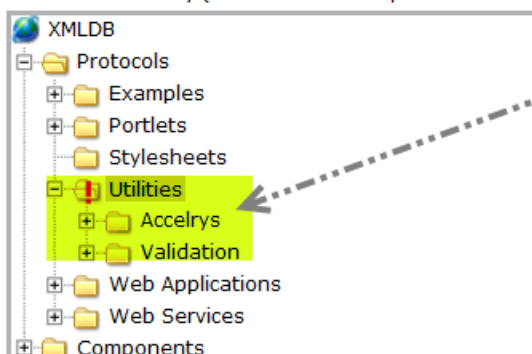
1. Go to **Security > Access Rights**.
2. Select the folder for which you want to restrict access.
3. To add a user or group to the access control list for this folder:
 - a. Select the type (*User or Group*).
 - b. Type the entity's name.
 - c. Select a permission (*Read/Write or Read Only*).
 - d. Click **Add/Edit**.

Examples

In this first example, the user ("User B") is assigned *Read/Write* permission for the folder "Utilities". No other users can access the selected folder.

Select Folder

Assign limited rights by selecting subfolders at different levels in the tab hierarchy (restricts access to specific folders only).



Access Permissions

Assign access rights for user and group entities.

| Entity | Type | Access Level | Remove |
|-----------|------|--------------|--------|
| User B | User | read/write | |
| everybody | | none | |

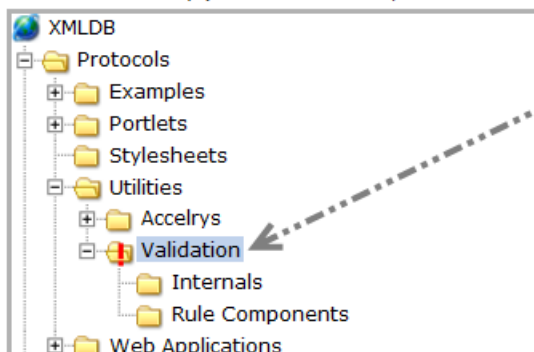
| Type | Entity Name |
|---------------------------------------------------------------------------------------------|----------------------|
| User | <input type="text"/> |
| Permission: | |
| <input type="radio"/> Read/Write <input type="radio"/> Read Only <input type="radio"/> None | |
| <input type="button" value="Add/Edit"/> | |

User with Read/Write permissions for a folder

In this second example, an entity group "ValidationUsers" is assigned *Read/Write* permission for the folder "Validation". Other users (*everybody*) have *Read* access to the same folder.

Select Folder

Assign limited rights by selecting subfolders at different levels in the tab hierarchy (restricts access to specific folders only).

**Access Permissions**

Assign access rights for user and group entities.

| Entity | Type | Access Level | Remove |
|-----------------|-------|--------------|--------|
| ValidationUsers | Group | read/write | |
| everybody | | read | |

| Type | Entity Name |
|----------------------|-------------|
| <input type="text"/> | everybody |

Permission:

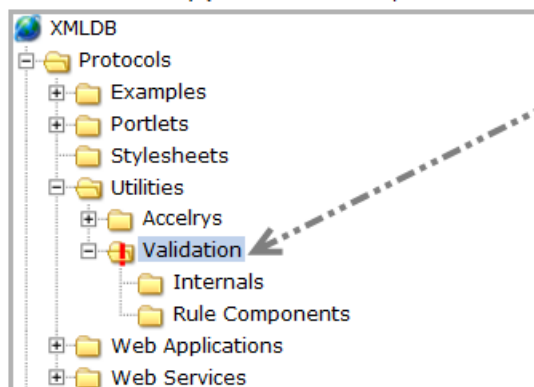
☐ Read/Write
 ☒ Read Only
 ☐ None

Entity with *Read/Write* permission for a folder

In this last example, the "ValidationTesters" group is assigned *Read/Write* permission for the "Validation" folder. The *everybody* entity has its permission set to *None*. Only members of the ValidationTesters group can access the folder.

Select Folder

Assign limited rights by selecting subfolders at different levels in the tab hierarchy (restricts access to specific folders only).

**Access Permissions**

Assign access rights for user and group entities.

| Entity | Type | Access Level | Remove |
|-----------------|-------|--------------|--------|
| ValidationUsers | Group | read/write | |
| everybody | | none | |

| Type | Entity Name |
|------|----------------------|
| User | <input type="text"/> |

Permission:


☐ Read/Write
 ☐ Read Only
 ☐ None

Entity with *Read/Write* permission for a folder. The *everybody* entity is assigned the *None* permission on the same folder to restrict access.

Notes:

- Standard protocols and components installed with Pipeline Pilot cannot be removed from the shared tabs or modified in any way.
- Users can move, remove, and rename their folders and subfolders. These actions impact users with restricted permissions in affected subfolders in the hierarchy. For example, if "Folder A" is renamed to "Folder Z", all users with restricted permissions in "Folder A" lose their publishing rights. Only rename shared folders if some scenario makes it absolutely necessary (not recommended as a standard practice).
- If you are sharing an XMLDB across servers, assign tab publishing permissions on the XMLDB *hosting server*, not on the remote computational servers.

To remove permissions:

1. Go to **Security > Access Rights**.
2. Select the folder for which you want to remove a permission.
3. From the **Access Permissions** list, select the entity name to highlight it.
4. Click **Remove** .

To modify permissions:

1. Go to **Security > Access Rights**.
2. Select the folder for which you want to change an entity.
3. From the **Access Permissions** list, select the entity name to highlight it.
4. The current permission is pre-selected. Select the new permission you want to assign the entity.
5. Click **Add/Edit**.

Notes:

- If a user does not have read access to a folder, it will not be visible in that user's Explorer window tab (i.e., protocols in that folder and its subfolders are not visible).
- You cannot assign *Read Only* permission to users or groups for the top-level Protocols or Components folders.

Impersonation and File Access Rights

This section concerns files in the file system of the Pipeline Pilot Server. When [Impersonation](#) is off, users can access and modify files based on the following settings:

- **Unrestricted File Browsing:** When enabled, users can read any file on the server.
- **Unrestricted File Editing:** When enabled, users can edit any file on the server.

Disabling one of the unrestricted file settings limits the actions (browsing or editing) to a user's user directory. If Impersonation is enabled, the operating system settings control a user's file access privileges and the values of these other two settings are ignored.

Authorizing Client Administrators

A *Platform/PipelinePilot/Administer* permission is available for Pipeline Pilot Server users. (Previously, this was a role called "PPClient Administrator".)


Client users with this permission are granted access rights to protocols stored in the "User Name" tabs for all other Pipeline Pilot users. This provides a way to manage protocols and components in the XMLDB and keep it well maintained for current users.

For example, if User A leaves a company, User B can easily take ownership of User A's protocols if he has client administrator privileges. User B can access User A's files from his desktop by enabling "Administrator Mode" in Pipeline Pilot (a right-click command available from the Network tab).

There is no need for User B to physically start Pipeline Pilot on User A's client. User B can claim ownership of the files for version tracking purposes, and he can also copy and move the files to other protocol databases. (For further details, see Pipeline Pilot Help.)

IMPORTANT! By default, the *Platform/PipelinePilot/Administer* permission is assigned to the *Platform/PowerUsers* group, but there are no members. An administrator must grant membership in this *PowerUsers* group to individual users or groups.

To assign members to the Platform/PowerUsers group:

1. Go to **Security > Groups**. The Manage Groups page opens.
2. Locate the *Platform/PowerUsers* group and then click **Edit** . The Group Assignment Editor opens for *Platform/PowerUsers*.
3. Assign members in the **Users** and **Groups** tabs.
4. Click **Continue** to close the editor and return to the Group Assignment table.
5. Click **Save** to save your changes.

Note: For these changes to go into effect, end users should restart their clients.

Authentication

Authentication Overview

Authentication allows you to control access to your server by verifying the identity of your network users based on their login names and passwords.

The following topics are covered:

- [Authentication Method](#)
- [SAML Web SSO Settings](#)
- [Anonymous Access](#)
- [Kerberos via SPNEGO](#)
- [Passing Insecure Passwords](#)
- [User Directory Sharing](#)
- [Restricting Permissions](#)
- [Notification Protocols](#)

Authentication Method

Pipeline Pilot offers several ways to authenticate, based on your organization's security policies and how you want to control access. Supported authentication methods include:

- **A list of users:** Administrators can define a list of users and passwords that can be used to log onto the server. If this is the only authentication method enabled, only the defined user names can be used to authenticate on this server.
This method can also be enabled in addition to one of the other methods listed below. In this case, the **User List** usually holds a small number of user names that do not have to exist in the corporate directory. These accounts could be used for specific roles such as the Administrator or as a "utility" user name for [anonymous access](#).
- **An external user directory:** This category supports authentication user accounts that are not maintained by Pipeline Pilot itself. Typically, this would be configured to use the corporate user directory. The options in this category can be used in addition to the Pipeline Pilot-maintained **User List**, if desired. The options are:
 - **Any User Name:** This option effectively disables authentication, since any entered user name is accepted as valid. The identity of the user is still important in terms of job ownership and permissions, but the identity has not been verified in any way. This option is enabled by default, but is replaced once you configure authentication based on a more restrictive method.

- **Local:** Refers to local accounts on the server operating system.
 - On Windows servers, this targets user accounts defined by the server Administrator on the machine itself, and not domain accounts.
 - On Linux servers, **Local** authentication is delegated to the Pluggable Authentication Modules (PAM), which can be configured to support multiple authentication architectures. By default, PAM supports authentication for users who have login access to the server operation system. For details, see [Security for Linux Authentication](#).
- **Domain:** This option is relevant only to a Windows server, and enables authentication of user credentials against a named Windows domain.
- **Foundation:** Foundation Hub will be used as the authentication server. See [Setting Foundation Hub as the Authentication Server](#).

IMPORTANT! If you have set Foundation Hub to be your Authentication server, you will manage your groups, users, and permissions from Foundation Hub. The **Security > Users**, **Security > Groups**, and **Security > Permissions** pages will not be available in the Pipeline Pilot Admin Portal.

- **3DPassport:** 3DPassport is Dassault Systèmes' user authentication server for the 3DS Platform. Setting up Pipeline Pilot for 3DPassport Authentication delegates the authentication to a 3DPassport server.
 - **Accept passwords via SSL only:** Reject passwords not passed via HTTPS.
 - **3DPlatform Service:** URL to the 3DPassport service.

Note: With all available authentication methods, the server creates a session for the user that includes the permissions defined for that user.

Setting Foundation Hub as the Authentication Server

IMPORTANT!

If Pipeline Pilot is installed on Linux, the Pipeline Pilot server hostname be a fully qualified domain name to successfully register with Foundation Hub. If the Linux server only has a short name, do the following:

1. Navigate to **Reverse Proxy and Load Balancing**.
 2. Set the **Reverse Proxy Name**. Set **Full Name** field to the fully qualified domain name.
 3. Set the **Reverse Proxy Ports** for **HTTP** and **SSL**.
1. Navigate to **Security > Authentication**.
 2. Set **Check users against** to **An external user directory**, and choose **Foundation**.
 3. Set **Foundation Server URL** to the fully qualified Foundation Hub server domain (load balanced endpoint if you are setting up a load balanced environment). For example: `https://<hub server name>.com:9953/foundation/hub`.
 4. Enter credentials for the **Foundation Admin**: `scitegicadmin/scitegic` by default.
 5. Set Impersonation:
 - **Full:** Processes spawned by protocol execution inherit the user credentials from the job process. This option requires specific Windows Local Security Policy settings.
 - **Restricted:** Processes spawned by protocol execution revert to the Apache user credentials. This option requires fewer Windows Local Security Policy settings.

- **None:** Processes spawned by protocol execution and their child processes revert to the Apache user credentials.

For further details, see [Managing Impersonation](#).

IMPORTANT! To use Impersonation for Foundation Hub Authentication, you must enable the following setting in Foundation Hub: **Foundation Hub Admin and Settings > Settings > Applications > Application Settings > Foundation Hub > Security > Save user password for duration of session**.

6. Click **Save**. After a few moments, the page will indicate a successful registration.

See the Foundation Hub Admin Guide for information about managing security once you have set the Foundation Hub to manage authentication.

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

Connecting Foundation Hub to Multiple Pipeline Pilot Servers

1. Configure the first Pipeline Pilot Server per the preceding instructions.
2. Repeat for each of the other servers. Be sure to use the same Foundation Hub URL when configuring each.
3. Open the Foundation Hub landing page.
4. Click the **Applications** icon at the top of the page, and click **Admin and Settings**.
5. Open **Settings > Applications**.
6. Check that all the Pipeline Pilot instances are visible and their Installation URLs are correct.

Update Registered Applications

You can use the **Foundation Applications** report in Pipeline Pilot to view the list of registered Foundation applications and update it manually if needed. Some BIOVIA applications are detected and registered automatically. However, if Pipeline Pilot packages are modified as part of an application installation and you will need to register them manually.

1. In Pipeline Pilot, navigate to **Reports > Foundation Applications** to check the list of registered applications.
2. Click **Update Applications** at the top-right corner of the table.

Registration Error

You may encounter the following error during registration:

Registration failed. Invalid server root URL [http://hostname:9944]. The server must register with its fully qualified URL. Example:
http://hostname.example.org:port Status: 422

This message indicates that the server was unable to determine its fully qualified server name in the network. To fix this:

1. Navigate to **Setup > Reverse Proxy and Load Balancing**.
2. Set **Full Name** to the fully qualified server name and set the **Reverse Proxy Ports** to the server's ports.

Disabling Authentication

By default, authentication is not enabled. A login process is not required and all clients can access the server.

To disable authentication:

1. Go to **Security > Authentication**.
2. Select "Any User Name" as the **Authentication Method**.

Note: Users with administrator permissions on the server are not allowed to access the Admin Portal unless authentication is enabled.

User List Authentication

User List authentication is practical when you have a limited number of users who require access, or in combination with corporate directory authentication, when you need to define additional user names that are not tied to corporate accounts.

IMPORTANT!

- If you are an enterprise user of BIOVIA Foundation, you will manage your groups and users from the MDM server and use the MDM Registration and Settings page in the Pipeline Pilot Server admin portal to associate it with the Pipeline Pilot Server and schedule synchronization of user, group, and permission data. Once you have registered with the MDM server, the users (and groups and permissions) functionality will not be available in the Pipeline Pilot Server admin portal.
- After you configure the server to support User List-based authentication, set up your list of users in Security > Users. The user names are written to a file on the server called "AuthUsers.xml". See [Managing Users for File-based Authentication](#).

To configure User List authentication:

1. First ensure that the User List is configured as required. The number of users currently defined is reported on the Authentication page.
2. Click the **Edit Users** button to manage the User List.
3. After setting up users, check **A list of users**.
4. Click **Save**.

Support for User List Authentication in Combination with Other Methods

You can employ **User List** authentication in conjunction with other authentication methods (e.g., Any User, Domain). Under this mixed authentication scenario, the usernames and passwords are checked against the User List *before* using the other method.

To use file-based authentication with another authentication method:

1. Check both **A list of users** and **An external user directory**.
2. Select the user directory method (e.g. Domain).

Tips:

- Ensure that the user names you define in the **User List** do not conflict with user names from the external directory (because **User List** account authentication takes priority, domain accounts are not able to log onto the server).
- If **Any User Name** is the authentication method, it is not necessary to include a password when logging on, provided the user name does not match any entry in the **User List**. If the user name is included in this list, the correct password must be provided.
- When using the Pipeline Pilot Client with a server with **Any User Name** authentication configured in addition to a **User List**, no logon prompt is displayed at startup (since the Windows user name is automatically used to log onto the server with no password to make the process as simple as possible for most users). **User List** user names that could clash with Windows login names should be avoided.
- Impersonation may be enabled with a mixed authentication configuration, but will only apply to user names that represent operating system users and not the user names in the **User List**. Any jobs executed by users authenticated in the **User List** can never support impersonation since the account does not exist on the host operating system.

Any User Name Authentication

By default, authentication is not enabled. A login process is not required and all user names can access the server. In this mode, the external user directory is set to **Any User Name**.

To reset the server to this non-authenticating state:

1. Go to **Security > Authentication**.
2. Check **An external user directory**.
3. Select "Any User Name" as the **Authentication Method**.

Note: Regardless of the authentication method, only those users with administrator permissions on the server are allowed to access the Admin Portal. By default, this means users that are members of the *Platform/Administrators* group.

Local Authentication

The Local authentication method leverages mechanisms on the local operating system to validate user credentials. Depending on the host operating system and network setup, the user accounts could exist solely on the local machine or originate from a corporate directory. If the user can log onto the server machine over native access mechanisms, Pipeline Pilot can be configured to authenticate against the user's account. However, user accounts in a corporate directory that do not have access to the host server will be denied access to Pipeline Pilot.

Requirements:

- **Windows:** For users to log onto Pipeline Pilot, they need to have accounts on the server. To set up local user accounts in Windows, open the Control Panel, navigate to the User Accounts section, and select "Give other users access to this computer". Details may vary on different versions of Windows.
- **Linux:** Authentication is handled with PAM. If you installed BIOVIA with root privileges, a default PAM configuration file is created ("/etc/pam.d/scitegic"). The default settings in this configuration file should be suitable for most situations. You can review the file and customize it as necessary. For further information, see [Security for Linux Authentication](#).

To configure local authentication:

1. Go to **Security > Authentication**.
2. Select **An external user directory**.
3. Select "Local" as the **Authentication Method**.
Relevant options are displayed in the **Settings** box.
4. Confirm the setting for **Accept passwords for SSL only** (for further details, see [Domain Authentication](#)).
5. Configure **Impersonation** for your operating system based on the following:
 - **Linux:** To enable impersonation, select the checkbox. When unchecked, impersonation is disabled and all jobs run as the user that is running the Apache HTTPD server.
 - **Windows:** To enable, set to "Full" or "Restricted". To disable, set to "None".
6. Decide if you need to [restrict permissions](#) to a select group of users.
7. Click **Save**.

Windows Impersonation

- **Full impersonation:** Inherits user credentials from any processes spawned by protocol execution. Requires specific Windows Local Security Policy settings.
- **Restricted impersonation:** Credentials revert to the Apache user in any process spawned during protocol execution, but require fewer settings.
- For further details, see [Managing Impersonation](#).

Domain Authentication

For a Pipeline Pilot installation on a Windows server, you can authenticate against one or more Windows domains. It is similar to the local authentication method, except that users are defined in the domain's Active Directory server.

To configure domain authentication:

1. Go to **Security > Authentication**.
2. Check **An external user directory**.
3. Select "Domain" as the **Authentication Method**.
4. In **Default Domain**, enter the domain names, separated by commas, based on the following:
 - To authenticate against a specific domain, enter a username that contains the domain name, followed by a backslash, (e.g., "DomainName\UserName").
 - When no domain is specified with the user name, the name is checked against the domains listed in the Default Domain(s) list.

Once successfully authenticated, the user's domain is cached. The next time the user attempts to authenticate, the cached domain will be used exclusively to minimize overhead.

5. Check **Allow SPNEGO (Kerberos)**, if required. (For further details on single sign on authentication with Kerberos, see [Support for Kerberos via SPNEGO](#).)
6. Confirm the setting for **Accept passwords for SSL only** (see Support for [Passing Insecure Passwords](#)).
7. Configure **Impersonation** as follows:
 - To enable impersonation, set to "Full" or "Restricted".
 - To disable impersonation, set to "None".

8. Decide if you need to restrict permissions to a select group of users (see [Restricting Permissions](#)).
9. Click **Save**.

Tips:

- If your Pipeline Pilot server moves to a different network domain, your users may need to specify the new domain when logging in (for example, MYDOMAIN\Username). Alternatively, you could specify the new domain explicitly in the **Default Domain** box on the **Authentication** page in the Pipeline Pilot Admin Portal.
- It is possible for users to select a domain name not specified in the Admin Portal. You can restrict access to only those domains you list in **Default Domain(s)**, by checking **Allow Domain access only from listed domains**.
- If a user does not specify a domain name, the server uses the domains in the order they are listed in the Admin Portal.
- Client user and domain names do not have to match Windows login names.

SAML Web SSO Settings

You can choose to Federate with an Identity Provider using SAML Web SSO. See [SAML Web SSO](#).

Anonymous Access

As an administration option, Pipeline Pilot supports anonymous execution of Pipeline Pilot protocols. Under anonymous execution, protocols launched anonymously will run as the user account specified in Anonymous Access.

To configure the server to support anonymous access:

1. Go to **Security > Authentication**.
2. In the Anonymous Access section, enter a **Username** and **Password** to authenticate any user who has is not currently logged in with a valid session cookie.

Notes:

- As an example, with anonymous access enabled, any web browser user without a valid current session who clicks a reporting link is automatically logged on, running as the anonymous user.
- The anonymous user credentials provided must be valid with the type of authentication configured on your server. If authentication is set to "Any User Name", there are no restrictions on the user name. Under any other authentication method, the username and password must be a valid login.
- Leaving the username blank disables anonymous job execution.
- It can be challenging to have a formal directory user name defined for use as the *Anonymous User* when it does not represent a single individual. This scenario is an opportunity to employ *User List* authentication. You can define a non-domain "utility" user who is not a member of the domain and enter this user's credentials in the Anonymous Access option. In this case, ensure that authentication against the User List is enabled in addition to Domain or Local authentication.

Support for Kerberos via SPNEGO

Authentication into Pipeline Pilot with Kerberos is supported via Simple and Protected GSS-API Negotiation Mechanism (SPNEGO) over HTTP. This supports the Integrated Windows Authentication (IWA) single sign on scheme. Some configuration is required on the server and on the client's browser.

Using Kerberos with Pipeline Pilot requires clients that support SPNEGO, such as:

- **Web browsers:** Internet Explorer, Firefox, Chrome
- **SDKs:** .NET Client SDK, JavaScript Client SDK
- **Not supported:** Other SDKs (Java) or Pipeline Pilot Client

Support for Kerberos on Windows

Pipeline Pilot currently supports delegating your Kerberos credentials on Windows if your Kerberos realm (for example, Active Directory) is configured to trust your server for delegation and access to Network file resources. For all other uses (for example, connecting to a database or SOAP service), it is necessary to explicitly provide credentials.

To enable SPNEGO on Windows:

1. Select "Domain" as the **Authentication Method** and configure according to the above instructions (see [Domain Authentication](#)).
2. Set **Impersonation** as desired (Kerberos is supported for all impersonation settings).
3. Check **Allow SPNEGO (Kerberos)**.

Support for Kerberos on Linux

To enable SPNEGO on Linux:

1. Select "Local" as the **Authentication Method** and configure per the above instructions.
2. Uncheck **Impersonation**.
3. Check **Allow SPNEGO (Kerberos)**.
4. (Optional) By default, Kerberos uses the keytab specified in "krb5.conf". To use a custom keytab file, enter the path in **Kerberos Keytab**.
5. (Optional) To accept the Kerberos token as a specific principal, enter the principal name in **Service Principal Name**. The name should start with "HTTP/" followed by the server's fully qualified domain name.

Note: SPNEGO authentication is currently not supported on servers behind a reverse proxy (including load balancers).

Configuring the Apache Service User Name for Kerberos

To run the Apache service as a user:

1. To set up a Kerberos service principal name for the Pipeline Pilot Server, run the following on the Domain controller:
`setspn.exe -a http/SERVERNAME DOMAIN\apache_username`
2. On the Pipeline Pilot Server, use the Services tool and set the Logon User for the "Pipeline Pilot x.x.x (Httpd)" service.

Client Configuration

To use Internet Explorer:

1. Add the server as a trusted site (Tools > Internet Options > Security > Trusted Sites > Custom Level > User Authentication > Logon).
2. Select **Automatic logon with current user name and password**.
3. If your server is already part of the Local Intranet, select **Automatic logon only in Intranet zone**.

To configure Chrome:

- Internet Explorer configuration will also allow Chrome to work with SPNEGO authentication, since it uses the Windows settings.

To configure Firefox:

1. Browse to "about:config".
2. Note the filter at the top of the page to help you find specific settings.
3. Add the server names to the following preferences:
`network.negotiate-auth.trusted-uris`
`network.automatic-ntlm-auth.trusted-uris`

Passing Insecure Passwords

The Basic Access Authentication scheme can be used to transmit credentials from a user agent to the Pipeline Pilot Server. This scheme does not offer any protection against eavesdropping. It is insecure unless used over an encrypted channel such as SSL.

To protect user credentials, the server will not request or accept basic authentication credentials over an insecure connection by default. Basic credentials are only accepted from connections originating from the same server or ones that use SSL. HTTP connections without SSL originating from other computers are not allowed to use basic authentication.

To override this behavior and enable basic access authentication over any connection:

- Uncheck **Accept passwords via SSL only**.

User Directory Sharing

Linux administrators can specify how users access other client's files. This applies to user subfolders in the "public/users" directory. The following options are available:

- **Deny:** Users may not access any files in other user's directories. All access is denied.
- **Allow-read:** Users may read files from other user's directories.
- **Allow-all:** Users may read files from other user's directories and they may create or copy files to other user's directories. However, in most cases, existing files may not be changed because of the permissions on those files.

Restricting Permissions

Required Permissions: All users need to have *Platform/Logon* permissions to log onto the server.

By default, all users have the *Platform/Logon* permission, because that permission is assigned to the *Platform/Users* group, which in turn includes the *Platform/Everyone* group. However, you can also restrict server access to a select group of users by changing the membership of the *Platform/Users* group.

To restrict access:

1. Create a group for the users who require a server logon. Alternatively, you might already have a group defined (for example, Active Directory).
2. Add the new or existing group as a member of the *Platform/Users* group.
3. Remove the *Platform/Everyone* group from the *Platform/Users* group.

Do not remove *Platform/Logon* permission from the *Platform/Administrators* group.

Notification Protocols

There is a parameter, `NotificationProtocol`, on the Implementation tab that sends information about the protocol run such as Job ID, Job Status, etc. to a notification protocol that you have created to notify users when the run is stopped. The protocol will be run using anonymous credentials, or credentials specific to your protocol.

Enabling the Notification Protocol

Set Notification protocols to one of the following:

- **Use Anonymous Credentials:** Run the protocol using the Anonymous Access credentials on the Authentication page.
- **Use Notification Credentials:** Use the credentials that appear below this option when selected.
- **Disabled**

NotificationProtocol Parameters

`NotificationProtocol` specifies the name or component ID (guid) of a protocol stored in the server's protocol database that will be executed when the current job completes. The notification protocol will receive the following parameters that contain information about the job:

- **Notify_JobID:** Job id of the execution.
- **Notify_JobStatus:** Description of the result of the job execution.
- **Notify_JobStatusCode:** Status code that for the result of the job execution.
 - 5: Job was stopped by the client or administrator.
 - 6: Job completed normally with success.
 - 7: Job completed with an error.
 - 8: The process ID associated with the running job crashed or otherwise disappeared.
 - 9: Job failed to start.
- **Notify_ProtocolName:** Name of the protocol.
- **Notify_ProtocolPath:** Path of the protocol in the DB. This field can be blank for protocols that were launched without saving to the database.
- **Notify_ProtocolLogName:** Log name of the protocol. This is usually the same as `ProtocolName`, however this can be set by the client to a different name than the protocol.
- **Notify_RunHost:** Name of the node where the protocol executed.
- **Notify_Username:** User that ran the job.

Security for Linux Authentication

On Linux, Local authentication is handled with the Pluggable Authentication Modules (PAM).

PAM provides a centralized mechanism for authenticating all services. It applies to login, remote logins (telnet, rlogin, and rsh), FTP, Point-to-Point Protocol (PPP), and su, among others. It allows for limits on access of applications, limits of user access to specific time periods, alternate authentication methods, additional logging, and more.

Pipeline Pilot's use of PAM is controlled by a configuration file called `/etc/pam.d/scitegic`. A default version of this file is created as an option during installation, by running a boot script using root privileges, named `scirootinstall`. Though the default settings in this configuration file should be suitable for most situations, you can review the file and customize it as necessary. If it is necessary to use a different path for the PAM configuration file, please contact Dassault Systèmes Customer Support.

The default information for the Red Hat Linux installation in this file is shown below; the SUSE Linux version is slightly different. (You can modify it to suit your security requirements.)

```
Line 1: auth      required pam_nologin.so
Line 2: auth      required pam_securetty.so
Line 3: auth      required pam_env.so
Line 4: auth      required pam_stack.so    service=system-auth
Line 5: account   required pam_stack.so    service=system-auth
Line 6: session   required pam_stack.so    service=system-auth
```

- Lines 1–4: Specify authentication modules. These modules provide user information, such as a password. They can also set credentials and grant privileges. `system-auth` provides a set of centralized policies on Red Hat.
- Line 5: Specifies an account module that checks on various aspects of the user's account, such as password aging, and limits access to particular time periods or from particular locations. This module can also limit system access based on system resources.
- Line 6: Specifies a session module that is used to provide functions before and after session establishment. This includes setting up an environment, logging, etc.

| Module | Description |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| pam_nologin.so | Checks to see if the file <code>/etc/nologin</code> exists. If <code>nologin</code> does exist and the user is not root, authentication fails. |
| pam_securetty.so | If the user is logging in as root, ensures that the tty for the user is listed in the <code>/etc/securetty</code> file, if that file exists. |
| pam_env.so | Sets the environmental variables specified in <code>/etc/security/pam_env.conf</code> . |

The default configuration will authenticate users based on the operating system security settings, which could be a local passwd file, LDAP authentication, NIS authentication, etc. This is consistent with support for user impersonation on the server. However, if impersonation is not in force and not all Pipeline Pilot users can log into the server machine, you will need to modify the configuration defined in the Pipeline Pilot's PAM file to utilize different pluggable PAM modules, such as the `pam_ldap` module.

Linux Shadow Password Support

Linux supports an authentication configuration where passwords to user accounts are stored in a file named `/etc/shadow` instead of `/etc/passwd`. When using shadow passwords on a Linux server with authentication enabled, the server fails to authenticate any user other than the user ID that owns the Apache process, even though the correct password is provided. This happens because the `/etc/shadow` file is only readable by root (superuser). To work around this issue, follow the instructions below.

To authenticate shadow passwords on Linux:

1. Log in as root.
2. Change the group of the shadow password file to the Pipeline Pilot apache users group:
`chgrp ppgroup /etc/shadow`
3. Change the permissions of the shadow file to be group readable:
`chmod g+r /etc/shadow`

Note: For further information about shadow passwords and authentication on Linux, review the Linux-PAM library at the [Linux Kernel Archives](#).

Notes About Fingerprint Reader Authentication

Some versions of Red Hat Linux set up optional fingerprint reader authentication by default as part of the system authentication, which causes the fingerprint reading libraries to be run during each authentication through Linux PAM. Some versions of the fingerprint scanner library, `/lib64/security/pam_fprintd.so`, are unstable and can cause application crashes in multi threaded applications such as Pipeline Pilot, when validating users via username/password authentication. When configuring Pipeline Pilot for "Local" authentication on Linux based servers, the `scirootinstall` script sets up a Pipeline Pilot PAM configuration file in `/etc/pam.d/scitegic`. The default configuration file in turn points the configuration towards the standard OS authentication file located in `/etc/pam.d/system-auth`. When installing Pipeline Pilot to support Linux based username/password authentication, we recommend that you check the `system-auth` configuration and ensure that the fingerprint scanner is disabled as follows:

1. Log in as root or via `sudo`.
2. Edit `/etc/pam.d/system-auth` file and look for the following line: `auth sufficient pam_fprintd.so`.
 - If this line is not present, you do not need to take action.
 - If this line is present, but your server is not configured to utilize a fingerprint scanner as a form of authentication, then disable this form of authentication using the following command:

```
authconfig --disablefingerprint --update
```

If you need to run the `authconfig` command, double check the `/etc/pam.d/system-auth` file afterwards.

Linux Username Case Sensitivity

Because the file system on Linux is case-sensitive, the directory names created for users are also case-sensitive. If a user authenticates with a username that already exists as a user directory differing only by case, the username in the session will be adjusted to match the pre-existing username, to avoid a proliferation of similar username directories. (If there is more than one case-insensitive match, the first directory returned from the file system is used).

Managing a User List for Authentication

If your authentication method includes checking against a list of users defined by the Administrator, the server maintains a list of users in a file called `"AuthUsers.xml"`. You can add and remove users from this file in the Admin Portal (**Security > Users**). If User List authentication is switched on, user credentials are validated against the name and passwords you provide.

IMPORTANT! If you have set Foundation Hub to be your Authentication server, you will manage your groups, users, and permissions from Foundation Hub. The **Security > Users** page will not be available. See [Managing Authentication](#) to learn how to set the Foundation Hub as the authentication server. See the Foundation Hub Admin Guide for information about managing security once you have set the Foundation Hub to manage authentication.

Tip: To verify your current authentication method, go to **Security > Authentication** and ensure that in the **Authentication Method** panel, **A list of users** is checked.

To add a user:

1. Go to **Security > Users**.
2. Enter the user's name.

3. Enter the user's password (in both fields).
4. Click **Add**. The user name you entered is added to the Users list.

To remove a user:

1. From the Users list, click the name.
2. Click **Remove**. The user name is removed from the Users list.

To update a user's password:

1. From the Users list, click the name.
2. Enter the user's new password (in both fields).
3. Click **Add**.

Managing Impersonation

Impersonation provides running protocol jobs with exclusive user-based access to resources and, conversely, prevents excess permission from being granted to a single utility user to cover all resource access.

With impersonation enabled, a server can run protocols under the client's user account instead of the server account. Clients can then use their network security credentials, instead of the server account credentials, to access network resources with exactly the privileges assigned to their user's account.

A single server typically runs protocols for multiple users where each user requires access to private data. Impersonation allows access to personal data without exposing that data to everyone on the network.

By default, all protocol jobs run under the same account on the server – the account running the Apache server. Since the same account is used, regardless of the client user running the job, the job process may have different network and file access rights than those of the end user.

For example, without impersonation, a client user is not able to complete a protocol that opens a file from his or her private space on the file server. In this scenario, the protocol generates file access errors when it runs, because the Apache user does not have sufficient file access permissions.

As a solution, the server can run protocols under a client's user name instead of the server account name. This is known as client impersonation.

Windows Requirements for Impersonation

Microsoft added additional security patches to later versions of the Windows operating system, which changed the behavior of client impersonation. Therefore, when running Pipeline Pilot on Windows, the following forms of impersonation are supported:

- **Full:** Processes spawned by protocol execution inherit the user credentials from the job process. This option requires specific Windows Local Security Policy settings.
- **Restricted:** Processes spawned by protocol execution revert to the Apache user credentials. This option requires fewer Windows Local Security Policy settings.
- **None:** Processes spawned by protocol execution and their child processes revert to the Apache user credentials.

For further details, see [Managing Impersonation](#).

IMPORTANT! To use Impersonation for Foundation Hub Authentication, you must enable the following setting in Foundation Hub: **Foundation Hub Admin and Settings > Settings > Applications > Application Settings > Foundation Hub > Security > Save user password for duration of session**.

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, and users should be logged into that domain.
- For full impersonation, configure the Apache service to run under the local system account. Alternatively, 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.

Note: The way the server is configured depends on which of these approaches you pursue.

Linux Requirements for Impersonation

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

- The PAM configuration file `/etc/pam.d/sci tegi c` must be correctly configured. For details, see [Security for Linux Authentication](#).
- The `scii` file must be installed with effective user of root permissions (if you run the `scirootinstall` script as root, this will be set up by the installer). The permission setting for `scii` should be 4110.
- If you change the location of the public user directory, XMLDB, or Jobs directory, the new directories must be owned by the Pipeline Pilot Apache user, and the group ID must be set to the Pipeline Pilot Apache group. The new directories must also have the `setgid` bit enabled. The permissions for these directories must be 2775.
- For optimal security, Pipeline Pilot users should not be members of the Pipeline Pilot Apache group (this group is reserved for the Pipeline Pilot Apache user account only).
- When impersonation is enabled, users cannot write to Pipeline Pilot system folders. They only have *Write* access to their own user folders, folders for jobs they have created, and to the temporary folder. The best approach for users is to assume that they can only access data in their public user directories.
- Impersonation makes it possible for you to specify how public users folders can be accessed by other users. One of the following access options can be selected in the Admin Portal:
 - **Allow-all:** Users can access and create new files in other users' folders.
 - **Allow-read:** Users can only read files in other users' folders.
 - **Deny-all:** Users cannot access any files in other users' folders.

Tip: For most installations, *Allow-read* or *Deny-all* is sufficient.

Client User Requirements under Impersonation

Under impersonation, client users must have *Read*, *Write*, *Execute*, and *Directory Browse* access to the following subfolders in the `<install>` root:

- `<install>/xml db`
- `<install>/web/jobs`
- `<install>/public/users`
- `<install>/public/data`

- <install>/public/bin
- <install>/public/scripts
- <install>/bin (windows)
- <install>/linux_bin (Linux)

Note: Impersonation is not enabled by default. When using Local or Domain as your authentication method, impersonation can then be enabled.

Enabling Impersonation

The Admin Portal provides settings to manage impersonation. These settings are related to user authentication settings.

To enable impersonation on Windows:

1. Go to **Security > Authentication**.
2. For **Authentication Method**, select either "Domain" or "Local" as an external user directory.
3. For **Impersonation**, select either "Full" or "Restricted".
4. Click **Save**.

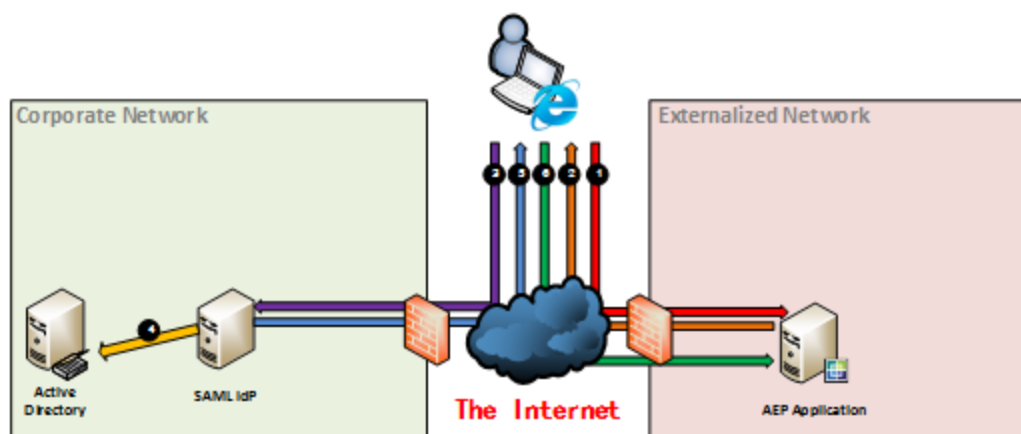
To enable impersonation on Linux:

1. Go to **Security > Authentication**.
2. For **Authentication Method**, select "Local" as an external user directory.
3. Check **Impersonation**.

SAML Web SSO

Single Sign-on (SSO) protocols allow users to authenticate on one system and transfer the authentication state to independent systems running other applications without needing to re-authenticate. Pipeline Pilot Servers can act as service providers (SP) in the authentication scenarios defined by the SAML 2.0 Web Browser SSO Profile specification. In these scenarios, an ecosystem comprised solely of Service Providers (SP) is not sufficient. Service providers rely on central Identity Providers (IDP) to perform initial authentication and manage user identity. The specification defines the steps that an SP and IDP must employ to securely exchange user identity. How the initial authentication used to identify the user to the IDP is left unspecified.

First-time Authentication



1. **Red:** The User Agent makes an unauthenticated request to a Pipeline Pilot application URL.
2. **Orange:** The Pipeline Pilot Service Provider sends an HTTP redirect to the User Agent that includes AuthnRequest.
3. **Purple:** The User Agent executes redirect and connects to the Identity Provider sending it the AuthnRequest.
4. **Yellow:** The Identity Provider obtains user credentials from the User Agent and validates them from the identity store.
5. **Blue:** The Identity Provider sends HTTP redirect to the User Agent which includes an authentication assertion.
6. **Green:** The User Agent enacts redirect and connects to Pipeline Pilot sending it an authentication assertion.

When web SSO is configured and enabled on a Pipeline Pilot Server, and a user makes a request without a valid session (1) Pipeline Pilot will initiate a handshake with the configured IDP (2) by instructing the browser to transmit an authentication request to the IDP (3). How the user authenticates against the IDP is not specified by the protocol, but can involve such technologies as HTML forms, two-factor authentication, or Kerberos tokens. After the user has validated with the IDP, the IDP instructs the user's browser to send an authentication response message to a service endpoint on the Pipeline Pilot Server (5). This endpoint will verify the authenticity of the message and ensure that its origin is the configured IDP (6).

SAML 2.0 services, whether acting as either identity providers or service providers, use XML documents to share metadata describing how other SAML services should interact with them. Metadata files contain service endpoint locations, details about supported features, and information to identify the providers themselves such as entity IDs and public keys. Typically metadata must be exchanged in both directions to establish a two-way trust relationship between an IDP and an SP before single sign-on authentications can occur.

Limitations

SAML Web SSO is only valid in interactions from a web browser, and is not recommended for headless clients, REST clients, thick clients, SOAP clients etc.

Configuring SAML Web SSO

To enable single sign on, Pipeline Pilot must be configured for SSO, and a two-way trust relationship between Pipeline Pilot and the IDP be defined before a user can use SAML for authentication. An administrator of the IDP must make changes to allow the Pipeline Pilot Service Provider to send authentication requests to the IDP.

Setting up SSO and SAML Certificates

1. Open the Server Home Page (<http://localhost:9943> be default).
2. Open **Security > SAML Certificate** and add a key pair and private key files. See *Admin > Security > Certificates > Managing SSO Certificates* in the help center for details.
3. Open **Security > SAML Certificate** and Set up a SAML Certificate. See *Admin > Security > Certificates > Managing SAML Certificates* in the help center for details.

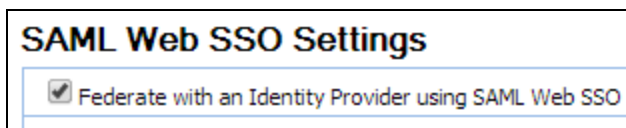
Note: Single Logout (SLO) will not work if you do not set up a SAML Certificate when the IDP requires a signed logout request message.

4. Restart the server.

Enabling SAML SSO and Configuring Service Provider Settings

The Authentication page on the Admin Portal has an option to enable or disable SAML Web Browser SSO.

1. Open **Security > Authentication**.
2. Choose **Federate with an Identity Provider (IDP) using SAML Web SSO**.



SAML Web SSO Settings

☒ Federate with an Identity Provider using SAML Web SSO

Note: When this option is not selected, the server will neither accept SAML credentials nor issue AuthnRequests to an IDP for unauthenticated requests. Users will rely on traditional authentication methods such as login form, HTTP Basic, or SP-NEGO to obtain a session.

3. Set the **Base URL**. This is the canonical URL that the IDP will use to contact Pipeline Pilot. Include the scheme, hostname and port number. Path information should be omitted (e.g., `http://pps.example.com:9944`).
4. Set the **Entity ID**. This used to identify this instance of Pipeline Pilot to the IDP. It should be a unique string that can be traced back to the host and service (e.g., `[Pipeline Pilot SAML2 web SSO]server.example.com:9943`).
5. Set **Request Signing** to **Sign Outbound AuthnRequests**. When a user initiates a web SSO login, Pipeline Pilot sends a user identity request to the IDP. Pipeline Pilot can sign the request to verify its integrity and origin.
6. Generate the metadata file for the IDP by clicking **Download**. The administrator of the IDP will load Pipeline Pilot's metadata and configure an SP connection. If you are the IDP administrator, refer to the IDP documentation.

Example

| | | |
|---------------------|-----------------------------------------------------------------|-------------------------------------|
| Base URL | <input type="text" value="https://spiral.accelrys.net:9943"/> | |
| Entity ID | <input type="text" value="saml-sp[spiral.accelrys.net_9943]"/> | |
| Request Signing | <input checked="" type="checkbox"/> Sign Outbound AuthnRequests | Manage Certificates |
| Metadata Generation | Download | |

Configuring Identity Provider Settings

After the IDP has the Pipeline Pilot SP metadata loaded and configured with a connection to the Pipeline Pilot SP, you will need to obtain IDP metadata from the admin and add it to the Identity Provider Settings.

The metadata could be published by the IDP at a public endpoint. It is also possible that the IDP administrator will need to generate metadata specifically for your Pipeline Pilot Server based on the SP connection settings defined in the IDP.

The metadata provides the details necessary for interacting with the IDP such as its EntityId, various IDP service endpoints, and its encryption and signing keys. The IDP metadata itself could be signed to ensure integrity during transport. If it is signed, then the Pipeline Pilot SP will attempt to validate the certificate path from the signing key to any trusted certificates in SAML Certificate store, which is managed under the Trusted Certificates tab of the SAML Certificates page in the Administration Portal. If certificate chain cannot be established to signing key of metadata, Pipeline Pilot SP will log an error and

not accept assertions from that IDP until it is resolved. You may alternately request unsigned metadata from your IDP if you believe that the metadata will not be altered in transit.

1. Obtain the IDP metadata file from the IDP administrator and paste the contents in the **Metadata** field.
2. Specify the **Group Attribute Name** adding a comma-separated list of SAML attribute names that hold group/role information. Pipeline Pilot will attempt to match these up to Pipeline Pilot Groups/Roles defined in the Groups page of the Admin Portal.
3. Save and restart the Pipeline Pilot Server again.

Example

| | |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Metadata | <pre><ds:keyinfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#" > <ds:X509Data> <ds:X509Certificate>MIICJTCCAY6gAwIBAgIGAULEN AtNMA0GCSqGSIb3DQEBBQUAMFYxCzAJBgNVBAYTA1VTMR EwDwYDVQQKEwhBY2N1bHJ5czEUMBIGA1UECXMRLGV2ZWx vcG11bnQxHjAcBgNVBAMTFWVzcHJlc3NvLmFjY2Vscnlz Lm5ldDAeFw0xMzEyMDUxOTIwMTRaFw0zMzExMzAxOTIwM TRaMEFYxCzAJBgNVBAYTA1VTMRFEwDwYDVQQKEwhBY2N1bH</pre> |
| Group Attribute Name | <input type="text" value="group, role"/> Manage Claims |

Verifying your Configuration

You can test this mode by trying to access `http[s]://<pps_host>:<pps_port>/webport/default/main.htm` from a browser. If you are not yet authenticated, the browser should be redirected to the IDP at this point. After authentication finishes on IDP, you will be automatically redirected to WebPort without any further authentication prompts or forms.

Hub Connection

Groups and Permissions

Groups and Permissions Overview

A group is a collection of users that define a role (e.g., platform administrators). A role is a permission to perform a task, (e.g., running Web Port, logging into the Admin Portal). Pipeline Pilot uses the term "permission" instead of "role" for this reason.

IMPORTANT! If you have set Foundation Hub to be your Authentication server, you will manage your groups, users, and permissions from Foundation Hub. The **Security > Groups** and **Security > Pages** will not be available. See [Managing Authentication](#) to learn how to set the Foundation Hub as the authentication server. See the Foundation Hub Admin Guide for information about managing security once you have set the Foundation Hub to manage authentication.

System Groups

The following system (default) groups are available with Pipeline Pilot:

| Group Name | Description |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Platform/Everyone</i> | All users implicitly belong to this group. Although there is no explicit assignment for this group, every user logged into Pipeline Pilot is a member. |
| <i>Platform/Users</i> | All general users for the Pipeline Pilot installation |
| <i>Platform/PowerUsers</i> | General user rights Pipeline Pilot Server administration user |
| <i>Platform/Administrators</i> | Admin Portal user Administration component user |
| <i>Platform/WebPort/Users</i> | Web Port user (this group is allowed <i>Platform/WebPort/Logon</i> permissions) |
| <i>Platform/DeniedUsers</i> | Prevents users from logging into the Pipeline Pilot (this group is denied <i>Platform/Logon</i> permissions). Add users to this group to conveniently deny them access to Pipeline Pilot. |

Note: By default, *Platform/Everyone* (every logged-in user) is a member of some of the other groups (e.g., *Platform/Users* and *Platform/WebPort/Users*).

Operating System Groups Only Define Group Membership

In previous releases of the platform software (i.e., Pipeline Pilot 8.5 and earlier), operating system groups could be used to define rights for the following:

- Access rights (XMLDB)
- Roles
- Data sources
- Group membership (Pipeline Pilot groups)

This is no longer the case. With Pipeline Pilot, operating system groups are only used to define group memberships. What this means:

- Groups defined in the Admin Portal are referred to as "Group" throughout the system (Admin Portal and components).
- To use operating system groups for defining rights, create a Pipeline Pilot group, assign it to the operating system group, and then define rights for that Pipeline Pilot group.
- Legacy migration is supported. For example, when the installer finds an operating system group used for a permission or access right, it will create a Pipeline Pilot group (Local_OsGroupName), assign it to the operating system group, and replace the permission/access right with this new group.

Pipeline Pilot System Permissions

The following default system permissions are available:

| Name | Permission to: |
|--------------------------------------|-------------------------------------------------------------|
| <i>Platform/Administration/Logon</i> | Log into the Admin Portal and use administration components |
| <i>Platform/PipelinePilot/Logon</i> | Log into Pipeline Pilot Server |

| Name | Permission to: |
|------------------------------------------|--------------------------------------------------------------------------------------------|
| <i>Platform/PipelinePilot/Administer</i> | Perform administration functions in Pipeline Pilot Server |
| <i>Platform/RunProtocol</i> | Use the RunProtocol command line tool |
| <i>Platform/WebPort/Logon</i> | Log into WebPort |
| <i>Platform/Logon</i> | Connect to the Pipeline Pilot server (needed to create a session and perform any function) |

Notes:

- Other BIOVIA-provided packages and collections can also define their own system permissions and groups. Depending on your particular installation of products, additional system permissions might be available.
- All system groups are assigned a set of Pipeline Pilot permissions by default.
- All users (*Platform/Users* group) have the *Platform/Logon* permission.
- Currently, there are a limited number of permissions. The *Platform/Administration/Logon* permission controls the entire Admin Portal.

Legacy vs. Pipeline Pilot System Permissions

Here is a comparison of legacy role names and new (renamed permissions) for Pipeline Pilot:

| Legacy Permission Name | Pipeline Pilot Permission Name |
|-------------------------------|------------------------------------------|
| <i>Admin Portal</i> | <i>Platform/Administration/Logon</i> |
| <i>PPClient</i> | <i>Platform/PipelinePilot/Logon</i> |
| <i>PPClient/Administrator</i> | <i>Platform/PipelinePilot/Administer</i> |
| <i>Run Protocol</i> | <i>Platform/RunProtocol</i> |
| <i>WebPort</i> | <i>Platform/WebPort/Logon</i> |
| ... | <i>Platform/Logon</i> |

Note: Previously, roles could be designated as "Allow All" (if no explicit assignment, all users had the role). Now, permissions must be explicitly assigned (if you haven't been assigned the permission, you do not have it).

Package-defined Permissions

Package-defined permissions are now supported. Each package can define the following:

- Groups
- Permissions
- Assignments

Groups

Groups have the following characteristics:

- Defined in `xml/objects/AuthGroups.xml`
- Include group members (e.g., *Platform/Everyone* is a member of *Platform/Users*). By default, all users are in *Platform/Users*, since all users are in its member *Platform/Everyone*.

Permissions

- Permissions are defined in `xml/objects/AuthPermissions.xml`.

Platform Permissions

- All platform groups, permissions, and assignments are defined in the `scitegic/generic` package.

Assignments

Assignments define how groups are allowed or denied permissions. Permission assignments can be overwritten by the administrator and these customizations are remembered when a package is reinstalled.

Package Permissions and Assignments

- All package permissions/groups have a namespace (`xxx/yyy`). For example, *Platform/Logon* where "Platform" is the namespace. The namespace is defined in the `package.conf` file.
- Group permission assignments are defined in `xml/objects/AuthAssignments.xml`.

Admin Portal Users

Previously, the Admin Portal had a different set of users to normal login users. In Pipeline Pilot, a consistent set of users has access to all platform functionality, but access is controlled by Permission assignments.

To make a user an Administrator:

- Add them directly (or indirectly) to the *Platform/Administrators* group.

Notes:

- By default, all users (*Platform/Everyone*) are administrators for a clean install.
- For migration of existing systems, only the users defined as Admin Users will be members of the *Platform/Administrators* group.

Custom Groups and Permissions

Creating custom groups is useful for several reasons:

- Administrators can use custom groups to define memberships.
- Memberships can then be assigned to resources (e.g., XMLDB folders).
- Memberships can also be assigned to system groups to define which users have access to the permissions associated to the system groups.

The purpose of custom permissions is to control access to user-defined protocols. For example, creating a custom permission is useful when a custom protocol includes a "Check User Has Permission" parameter for that custom permission.

Support for Groups and Permissions in Pipeline Pilot

Pipeline Pilot's security features are designed to check that the user has the appropriate permission to perform actions. When the user logs in, they are given a session, and this session caches the user's group membership and permissions.

Tip: To see what groups and permissions are assigned to a user, run a protocol in Pipeline Pilot that uses the *Output from the Current Session* component.

Some utility components are also available that can be used to detect groups and permissions:

- **Check User Is Group Member:** Passes the record out the Pass port if the user is a member (directly or not) of the specified group (i.e. the group is specified in the user's session). The record is passed out the Fail port if the user does not belong to the group.
- **Check User Has Permission:** Passes the record out the Pass port if the user has the specified permission (directly or through a group for which it is a member, i.e., the permission is specified in the user's session). The record is passed out the Fail port if the user does not have the permission.

Getting Started with Groups

Use the Manage Groups page (**Security > Groups**) to review and maintain groups and assignments.

IMPORTANT! If you have set Foundation Hub to be your Authentication server, you will manage your groups, users, and permissions from Foundation Hub. The **Security > Groups** page will not be available. See [Managing Authentication](#) to learn how to set the Foundation Hub as the authentication server. See the Foundation Hub Admin Guide for information about managing security once you have set the Foundation Hub to manage authentication.

Manage Groups

Use the following table to review and edit group membership and permission assignment for groups. The columns in the table display the number of members of different types assigned to the group as well as the number of permission allowed or denied to the group. For example, the numbers under "Membership Count/Groups" indicate the number of member groups associated with the group. Positioning the mouse over a number will reveal the underlying set of assigned members or permissions.

IMPORTANT! Group modifications will be reflected in the client after it has been restarted.

Current Group Assignment Table

Filter By Group:

| Delete | Group | Type | Edit | Reset | Current Mem | |
|--------|-------------------------|--------|------|-------|-------------|--|
| | | | | | Users | |
| | Platform/Administrators | System | | | 2 | |
| | Platform/DeniedUsers | System | | | 0 | |
| | Platform/PowerUsers | System | | | 0 | |

Group Management Features

Group Assignment Table

A Group Assignment table is available for managing groups and group members (users, groups, and external groups). Features include:

| Use this feature: | To do this: |
|--------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| Filter by Group <input type="text" value="Filter By Group: Search term ..."/> <input type="button" value="x"/> | When a long list of names is displayed, narrows down the list based on text you type in this field. |






| Use this feature: | To do this: |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Add Custom Group  | Adds a new custom group that you define. A name can contain any alphanumeric character except : ; / or \. |
| Save  | Saves your changes to the XMLDB. Notes: Notes: <ul style="list-style-type: none"> ■ Changes are displayed in red until you save them. ■ End users should restart their clients to see these changes. |
| Cancel  | Rolls back additions and modifications to the way they were before you saved the latest changes to the assignment table. |
| Detailed View <input type="checkbox"/> Detailed View | When checked, expands the table to display details about all assignments and permissions. |
| Delete <div> <div>Delete</div> <div>Group</div> <div> <input type="checkbox"/> Development Group Test <input checked="" type="checkbox"/> IT Group Test </div> </div> | Marks a custom group as an item to delete the next time you save changes. Items checked to delete are displayed in red until you save changes. |
| Edit  | Opens the Assignment Editor so you can assign members to groups. |
| Reset  | Resets changes you made in the Assignment Editor back to the way they were before you made the edits. (Hint: Like an undo feature.) |

Table Columns




| Column | Description |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Group | List of groups currently available on your server for which you can assign members. |
| Type | Identifies the type of group assigned. Custom groups that you add are identified as "Custom". Installed (default) groups are displayed as "System". |
| Current Membership Assignments | Number of members that belong to the selected group, organized by member type (Users, Groups, and External Groups). |
| Current Permission Assignments | Number of group members that are allowed or denied permissions. |

Tips:

- To quickly identify the users that belong to a specific group, point your mouse over a number in one of the columns (Users, Groups, External Groups). A tooltip displays details about the underlying set of assigned members.
- You can also check **Detailed View** to expand the table view with more details.

Group Assignment Editor

A Group Assignment Editor is available for managing group members. You can open this editor by clicking **Edit** for any group name listed in the group assignment table.

| Current Group Assignment Table | | | | | | |
|--------------------------------|-------------------------|-----------------|-----------------------------------------------------------------------------------|--------------------|-----------|-------|
| Filter By Group: | | Search term ... | ✕ | ➕ Add Custom Group | Save | ✕ Can |
| Delete | Group | Type ▲ | Edit | Reset | Current M | |
| | | | | | Users | |
| <input type="checkbox"/> | IT Test | Custom |  | 0 | 0 | |
| | Platform/Everyone | System |  | 0 | 0 | |
| | Platform/Administrators | System |  | | | |

Invoke editor for selected group

Group Assignment Editor Tabs

| Tab | Description |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Users | <ul style="list-style-type: none"> ■ Available Users: Displays a list of all users that are available to assign to the selected group. ■ Member Users: Displays a list of all users that are current members of the selected group. |

Group Assignment Editor: Platform/Administrators

Users Groups External Claims Permissions

Search term ... ✕

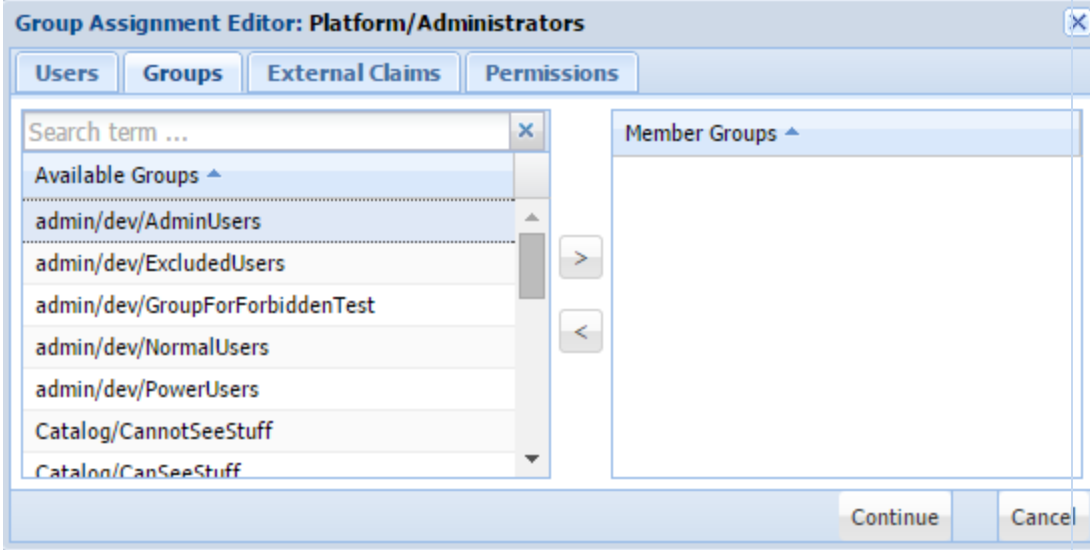
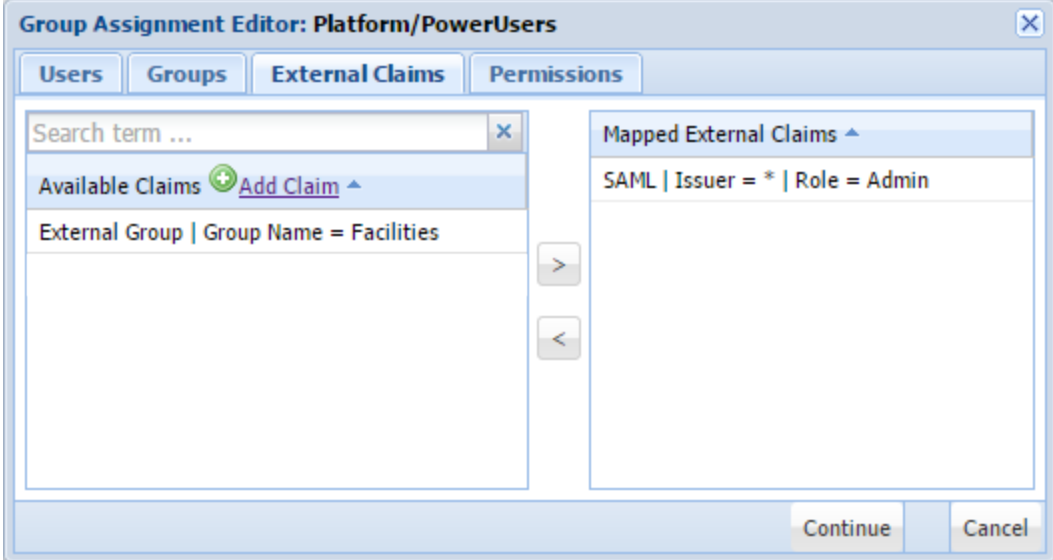
Available Users ➕ Add User ▲

admin
dwight
ehurd
hirobumi.kurosu
mdmLocalUser
ppuser

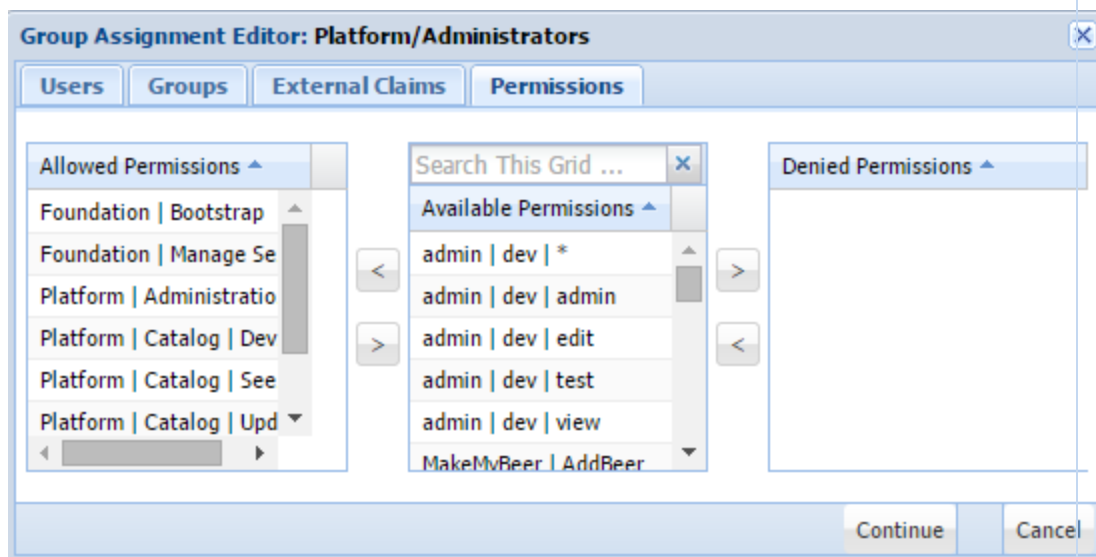
Member Users ▲

aknight
scitegicadmin

Continue Cancel

| Tab | Description |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Groups | <ul style="list-style-type: none"> ■ Available Groups: Displays a list of all groups that are available to assign to the selected group. ■ Member Groups: Displays a list of all groups that are current members of the selected group.  |
| External Claims | <p>An External Claim is an external piece of identity data provided by an identity provider. In addition to authenticating a user, the identity provider can specify one or more external claims associated to the user. Often they correspond to a local/system group, Domain group (Windows only), or a SAML claim or assertion.</p> <ul style="list-style-type: none"> ■ Available Claims: List of all available claims that can be mapped to the group. ■ Mapped External Claims: List of claims that are mapped the group.  |

| Tab | Description |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Permissions | <ul style="list-style-type: none"> ■ Allowed Permissions: Displays a list of all allowed permissions assigned to the selected group. ■ Available Permissions: Displays a list of all permission assignments that are available to assign to the selected group. ■ Denied Permissions: Displays a list of all denied permissions assigned to the selected group. |



Tip: When a long list of names is displayed in any column, use the **Search** box to narrow down the list based on text you type in this field.

Managing Custom Groups

Pipeline Pilot ships with a set of predefined/default groups called "system groups". Examples of system groups include *Platform/Administrators* and *Platform/Users*. (System groups all start with a namespace e.g. "*Platform/*").

In addition, a predefined set of permissions is associated to each system group. When an administrator is configuring Pipeline Pilot, it is common to create new custom groups unique to an organization. Typically, these groups define sets of individuals in the organization (either directly or through external groups) that have a particular function (e.g., chemists, informatics administrators).

The recommended approach to configuring the server is to assign your custom groups to one or more of the Pipeline Pilot system groups (e.g., assign a custom group called "Customer Informatics Administrators" to the *Platform/Administrators* and *Platform/PowerUsers* groups). As the set of desired users changes, you only need to update the membership of the appropriate custom group(s).

Note: Custom groups cannot use a namespace (i.e., the "/" character).


Follow these guidelines to support authorization on your server:

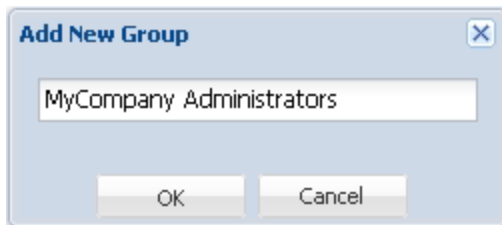
- To ensure that administrators can log into the Admin Portal, always leave *Platform/Logon* in the *Platform/Administrators* group.
- Assign permissions to *groups* (not users).

- Create custom groups that reflect an organization (e.g., *MyCompany Users*).
- Define membership in these groups through operating system groups (typically groups defined in an Active Directory) or by explicit user or group memberships.
- Assign these groups as members to the package-defined groups, (e.g., remove *Platform/Everyone* from *Platform/Users* and replace with *MyCompany Users*).

Adding a New Custom Group








To add a new custom group:

1. Go to **Security > Groups**. The Group Assignment table is displayed.
2. Click **Add Custom Group** .
3. In the Add New Group dialog, enter the name of your custom group.



A name can contain any alphanumeric character except : ; / | or \.

4. Click **OK**. The new group is added to the Group Assignment table. It is displayed in red until you save changes to the XMLDB.

| Current Group Assignment Table <input type="checkbox"/> Detailed View | | | | | | |
|----------------------------------------------------------------------------------------------------------|--------------------------|----------------------------------------------|------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|--------|
| Filter By Group: | | <input type="text" value="Search term ..."/> |  Add Custom Group |  Save |  Cancel | |
| Delete | Group | Type | Edit | Reset | Current Membership A | |
| | | | | | Users | Groups |
| <input type="checkbox"/> | MyCompany Administrators | Custom |  |  | 0 | 0 |
| <input type="checkbox"/> | Development Group Test | Custom |  | | 0 | 0 |
| | Platform/Administrators | System |  | | 1 | 1 |

5. Repeat the above steps to add additional custom groups.
6. Click **Save**. The custom group is added to the XMLDB.

Removing a Custom Group

To remove a custom group:

1. Check the box in the **Delete** column for the group you want to remove. When checked, the row is displayed in red to indicate that it's marked for deletion.

| Delete | Group |
|-------------------------------------|------------------------|
| <input type="checkbox"/> | Development Group Test |
| <input checked="" type="checkbox"/> | IT Group Test |




2. Repeat for each group to remove.
3. Click **Save**. The group is removed from the XMLDB.

Renaming a Custom Group

It is not possible to rename groups. Instead, delete the old group and then add a new group using the preferred name.

Managing Group Assignments




A Group Assignment Table is available for assigning members to groups (Security > Groups > Edit a selected group). Valid member types you can assign include users, system groups, custom groups, and external groups.

| Current Group Assignment Table | | | | | | |
|--------------------------------|-------------------------|-----------------|-----------------------------------------------------------------------------------|-------|-----------|--------|
| Filter By Group: | | Search term ... | + Add Custom Group | | Save | Cancel |
| Delete | Group | Type | Edit | Reset | Current M | |
| | | | | | Users | |
| <input type="checkbox"/> | IT Test | Custom |  | 0 | 0 | |
| | Platform/Everyone | System |  | 0 | 0 | |
| | Platform/Administrators | System |  | | | |

Invoke editor for selected group

Assigning Users to Groups


To assign a user to a group:


1. Go to **Security > Groups**. The Group Assignment table is displayed.
2. For the group you want to update, click **Edit** . The Group Assignment Editor opens.
3. Click the **Users** tab.
4. All users you can assign are listed in **Available Users**. From here, select one or more names. To assign multiple users at the same time, press and hold CTRL and click each name.
5. Click **Add to users** . The selected user names are moved to the Member Users list.
6. Click **Continue** to save your changes and close the editor. The Group Assignment table updates to show the newly added user members for the selected group. The changes are displayed in red until you save them to the XMLDB.
7. Click **Save**  to save your new member assignments to the XMLDB.


Note: For these changes to go into effect, end users should restart their clients.


Assigning Group Member Types

To assign a group member to a group:

1. Go to **Security > Groups**. The Group Assignment table is displayed.
2. For the group you want to update with group member types, click **Edit** . The Group Assignment Editor opens.
3. Click the **Groups** tab.
4. All group members you can assign are listed in **Available Groups**. From here, select one or more names. To assign multiple groups at the same time, press and hold CTRL and click each name.

- Click **Add to groups** . The selected group names are moved to the Member Groups list.



Tip: To remove a group member from the Member Groups list, select the name and then click **Remove from groups** . The name is moved back to the Available Groups list.

- Click **Continue** to save your changes and close the editor. The Group Assignment table updates to show the newly added group members for the selected group. The changes are displayed in red until you save them.
- Click **Save**  **Save** to save your new member assignments to the XMLDB.

Note: For these changes to go into effect, end users should restart their clients.

Mapping External Claims

To map external claims to a member group:

- Go to **Security > Groups**. The Group Assignment table is displayed.
- For the group you want to update with external claim types, click **Edit** . The Group Assignment Editor opens.
- Click the **External Claims** tab.
- All claims you can assign are listed in Available Claims. From here, select one or more names. To assign multiple claims at the same time, press and hold CTRL and click each name.
- Click the **Add to external claims** arrow. The selected claim names are moved to the Mapped External Claims list.
- To remove a claim from the Mapped External Claims list, select the name and then click the **Remove from external claims** arrow. The name is moved back to the Available Claims list.
- If the claim you want to map is not listed, click **Add Claim** to open the New External Claim dialog:
 - To add an External Group (local or domain), type the name of the group in the **External Group** tab.
 - To map a SAML claim, choose the **Assertion Issuer** (use * for any issuer), **Attribute Name**, and **Attribute Value** in the **SAML Assertion Attribute** tab. Note that this option is only available if the server is configured to federate with an Identity Provider using SAML Web SSO.
- Click **Continue** to save your changes and close the editor. The Group Assignment table updates to show the newly added external member groups.
- Click **Save**  **Save** to save your new member assignments to the XMLDB.

Getting Started with Permissions

Use the Manage Permissions page (**Security > Permissions**) to review and maintain permissions.

IMPORTANT! If you are an enterprise user of Biovia Foundation, you will manage your groups and users from the MDM server and use the MDM Registration and Settings page in the Pipeline Pilot Server admin portal to associate it with the Pipeline Pilot Server and schedule synchronization of user, group, and permission data. Once you have registered with the MDM server, the users, groups, and permissions functionality will not be available in the Pipeline Pilot Server admin portal. See MDM Server Configuration.

| Current Permission Assignment Table <input type="checkbox"/> Detailed View | | | | | | | | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|--------|------|-------|---------------------------|--------|--------------------------|--------|--------------------------------------------------------|
| Filter By Permission: <input type="text" value="Search term ..."/> + Add Custom Permission Save Cancel | | | | | | | | | |
| Delete | Permission | Type | Edit | Reset | Current Group Assignments | | Current User Assignments | | Description |
| | | | | | Allowed | Denied | Allowed | Denied | |
| | Platform Administration Logon | System | | | 1 | 0 | 0 | 0 | Permission to log on to the Administration Portal. |
| | Platform Catalog DeveloperLogon | System | | | 1 | 0 | 0 | 0 | Permission to use Solr administration tools. |
| | Platform Catalog SearchIndex | System | | | 1 | 0 | 0 | 0 | Permission to query the Accelrys Catalog index. |
| | Platform Catalog SeeAllIndex | System | | | 1 | 0 | 0 | 0 | Permission to see all entries in the Accelrys Catalog. |
| | Platform Catalog UpdateIndex | System | | | 1 | 0 | 0 | 0 | Permission to update the Accelrys Catalog index. |
| | Platform Logon | System | | | 3 | 1 | 0 | 0 | Permission to log on to the platform (get a session). |
| | Platform PipelinePilot Administer | System | | | 2 | 0 | 0 | 0 | Permission to administer Pipeline Pilot. |
| | Platform PipelinePilot Logon | System | | | 2 | 0 | 0 | 0 | Permission to log on to Pipeline Pilot. |
| | Platform RunProtocol | System | | | 3 | 0 | 0 | 0 | Permission to run a protocol using the RunProtocol. |
| | Platform WebPort Logon | System | | | 1 | 0 | 0 | 0 | Permission to log into Web Port. |
| | QueryService Administer | System | | | 1 | 0 | 0 | 0 | Query service administrator role. |
| | QueryService Logon | System | | | 1 | 0 | 0 | 0 | Query service regular user role. |

Permission Management Features

Permission Assignment Table

A Permission Assignment Table is available for managing permissions and permission assignments. Features include:

| Use this feature: | To do this: |
|--------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Filter by Permission Filter By Permission: <input type="text" value="Search term ..."/> | When a long list of permission names is displayed, narrows down the list based on text you type in this field. |
| Add Custom Permission <input type="button" value="+ Add Custom Permission"/> | Adds a new custom permission that you define. A name can contain any alphanumeric character except : ; / or \. |
| Save Save | Saves your changes to the XMLDB. Notes: Notes: <ul style="list-style-type: none"> ■ Unsaved changes are displayed in red before you click Save. ■ End users should restart their clients to see these changes. |
| Cancel Cancel | Rolls back additions and modifications to the way they were before you saved the latest changes to the assignment table. |
| Detailed View <input type="checkbox"/> Detailed View | When checked, expands the table to display details about all assignments and permissions. |
| Delete <div> <div>Delete</div> <div>Permission</div> </div> <div> <input type="checkbox"/> IT Test Administrators </div> | Marks a custom group as an item to delete the next time you save changes. Items checked to delete are displayed in red until you save changes. |








| Use this feature: | To do this: |
|---------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Edit  | Opens the Assignment Editor so you can assign permissions to groups. |
| Reset  | Resets changes you made in the Assignment Editor back to the way they were before you made the edits. (Hint: Like an undo feature.) |

Table Columns

| Column | Description |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Permission | List of permissions currently available on your server for which you can assign to group members. |
| Type | Custom permissions that you add are identified as "Custom". Installed (default) permissions are displayed as "System". |
| Current Group Assignments | Number of group members who are currently assigned the permission, organized by type (Allowed or Denied). |
| Current User Assignments | Number of user members who are currently assigned the permission, organized by type (Allowed or Denied). |
| Description | Provides further details about a selected permission. If you include details when you add a custom permission, the information is displayed in this column. |

Tips:

- To quickly identify the members that are currently allowed or denied permissions, point your mouse over a number in one of the columns (Allowed or Denied). A tooltip displays details about the underlying set of permissions.
- You can also check **Detailed View** to expand the table view with more details.

| Current Permission Assignment Table | | | | | | | | Detailed View |
|--------------------------------------------------------------------|-------------------------------------|-------------------------|-------------------------------------------------------------------------------------|-------|---------------------------|--------|--------------------------|---------------|
| Filter By Permission: <input type="text" value="Search term ..."/> | | + Add Custom Permission | | Save | Cancel | | | |
| Delete | Permission | Type | Edit | Reset | Current Group Assignments | | Current User Assignments | |
| | | | | | Allowed | Denied | Allowed | Denied |
| | Platform Administration Logon | System |  | | 1 | 0 | 0 | 0 |
| | Platform Catalog DeveloperLogon | System |  | | 1 | 0 | 0 | 0 |
| | Platform Catalog SearchIndex | System |  | | 1 | 0 | 0 | 0 |
| | Platform Catalog SeeAllIndex | System |  | | 1 | 0 | 0 | 0 |
| | Platform Catalog UpdateIndex | System |  | | 1 | 0 | 0 | 0 |

Allowed Groups:
Platform/Administrators

Permission Assignment Editor

A Permission Assignment Editor is available for permission assignments. You can open this editor by clicking Edit for any permission name listed in the permission assignment table.

| Current Permission Assignment Table | | | | | | |
|-------------------------------------|-------------------------------------|-----------------|------------------------------------------------------------|-------|--------------------------|--------|
| Filter By Permission: | | Search term ... | Add Custom Permission Save | | | |
| Delete | Permission | Type ▲ | Edit | Reset | Current Group Assignment | |
| | | | | | Allowed | Denied |
| <input type="checkbox"/> | MyCompany Administration | Custom | | | 0 | 0 |
| | Platform Administration Logon | System | | | | |
| | Platform Catalog DeveloperLogon | System | | | 1 | 0 |

Invoke editor for selected

Permission Assignment Editor Tabs

| Tab | Description |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Users | <ul style="list-style-type: none"> ■ Allowed Users: Displays a list of all users that are currently allowed the permission. ■ Available Users: Displays a list of all users that can be allowed or denied the permission. ■ Denied Users: Displays a list of all users that are currently denied the permission. |

Permission Assignment Editor

Users | Groups

Allowed Users ▲

bdreyer

Search This Grid ...

Available Users [Add User](#)

ytang

stest4

scitegicadmin

ppuser

keith.taylor

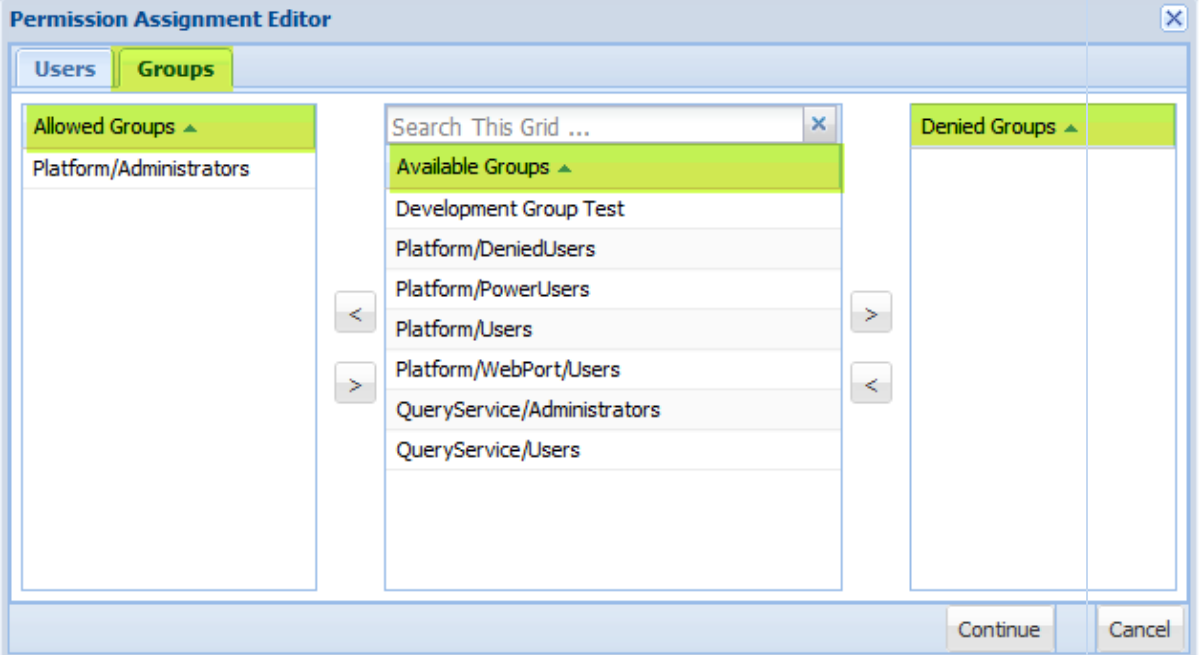
jku

dhoneycutt

Denied Users ▲

jschmoe

Continue Cancel

| Tab | Description |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Groups | <ul style="list-style-type: none"> ■ Allowed Groups: Displays a list of all member groups that are currently allowed the permission. Available Groups: Displays a list of all member groups that can be allowed or denied the permission. Denied Groups: Displays a list of all member groups that are currently denied the permission.  |


Tip: When a long list of names is displayed in any column, use the **Search** box to narrow down the list based on text you type in this field.

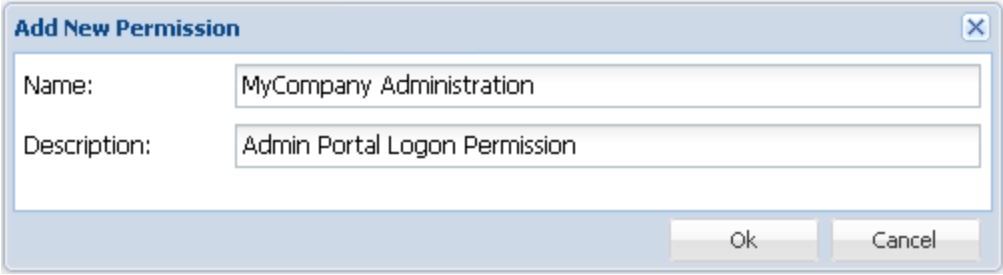
Managing Custom Permissions

Adding a New Custom Permission

A Custom Permissions page is available for creating new permissions and modifying existing permissions (Security > Custom Permissions).

To add a new custom permission:

1. Go to **Security > Permissions**. The Permission Assignment Table is displayed.
2. Click **Add Custom Permission** .
3. In the Add New Permission dialog, enter the name of your custom permission. Be sure to type a brief description so you can remember the purpose of the permission later on when you look it up in the Permission Assignment Table.



Add New Permission

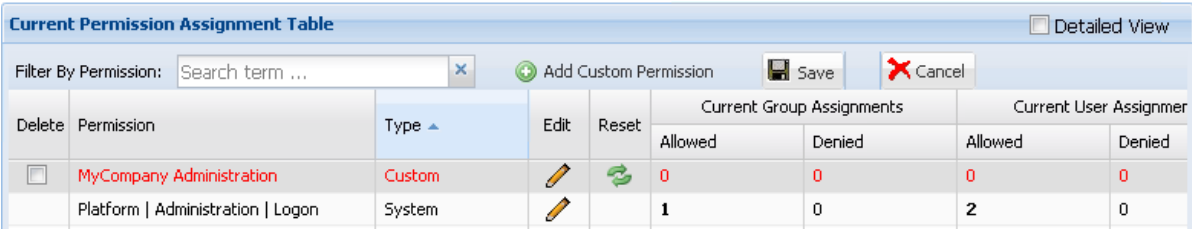
Name: MyCompany Administration

Description: Admin Portal Logon Permission

Ok Cancel

A name can contain any alphanumeric character except : ; / | or \.

- Click **OK**. The new permission is added to the Permission Assignment Table. It is displayed in red until you save changes to the XMLDB.



Current Permission Assignment Table ▢ Detailed View

Filter By Permission: Search term ... + Add Custom Permission Save Cancel

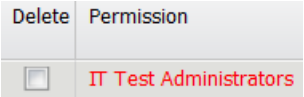
| Delete | Permission | Type | Edit | Reset | Current Group Assignments | | Current User Assignments | |
|--------------------------|-----------------------------------|--------|------|-------|---------------------------|--------|--------------------------|--------|
| | | | | | Allowed | Denied | Allowed | Denied |
| <input type="checkbox"/> | MyCompany Administration | Custom | | | 0 | 0 | 0 | 0 |
| | Platform Administration Logon | System | | | 1 | 0 | 2 | 0 |

- Repeat the above steps to add additional custom permissions.
- Click **Save**. The custom permissions are added to the XMLDB.

Removing a Custom Permission

To remove a custom permission:

- Check the box in the Delete column for the permission you want to remove. When checked, the row is displayed in red to indicate that it's marked for deletion.



| Delete | Permission |
|-------------------------------------|------------------------|
| <input checked="" type="checkbox"/> | IT Test Administrators |

- Repeat for each permission to remove.
- Click **Save**. The permission is removed from the XMLDB.

Renaming a Custom Permission

It is not possible to rename permissions. Instead, delete the old permission and add the new one using the preferred name.

Managing Permission Assignments

A Permission Assignment Table is available for assigning permissions to users and groups (Security > Permissions > Edit a selected permission). Permissions are either allowed or denied a user or group.

| Current Permission Assignment Table | | | | | | |
|--------------------------------------------------------------------|-------------------------------------|---------------------------------------|------|----------------------|--------------------------|--------|
| Filter By Permission: <input type="text" value="Search term ..."/> | | Add Custom Permission | | Save | | |
| Delete | Permission | Type ▲ | Edit | Reset | Current Group Assignment | |
| | | | | | Allowed | Denied |
| <input checked="" type="checkbox"/> | MyCompany Administration | Custom | | | 0 | 0 |
| | Platform Administration Logon | System | | | | |
| | Platform Catalog DeveloperLogon | System | | | 1 | 0 |

Invoke editor for selected

Assigning Group Permissions

To assign group permissions:

1. Go to **Security > Permissions**. The Permission Assignment Table is displayed.
2. For the permission you want to assign click **Edit** . The Permission Assignment Editor opens.
3. Click the **Groups** tab.
4. From **Available Groups**, select the group name and specify the permission:
 - To allow the permission for the group, click **Add to allowed list** . The selected group names are added to the Allowed Groups list on the left.
 - To deny the permission for the group, click **Add to denied list** . The selected group names are added to the Denied Groups list on the right.

Permission Assignment Editor: MyCompany Administration

Users **Groups**

Allowed Groups ▲

Available Groups ▲

- Development Group Test
- MyCompany Administrators
- Platform/Administrators
- Platform/DeniedUsers
- Platform/Users
- Platform/WebPort/Users
- QueryService/Administrators
- QueryService/Users

Denied Groups ▲

[Continue](#) [Cancel](#)




Tip: To remove a group from the allowed or denied list, select the name and then click either **Remove from Allowed List** or **Remove from Denied List** . The name is moved back to the Available Groups list.

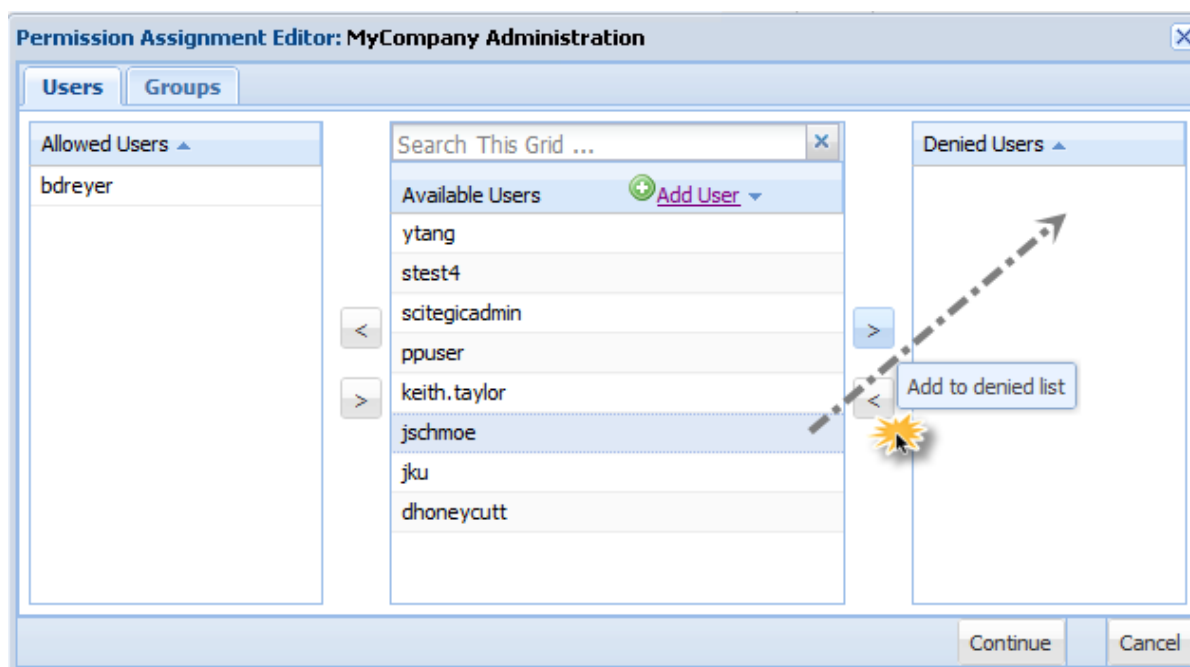
- Click **Continue** to save your changes and close the editor. The Permissions Assignment Table updates to show the newly assigned group permissions.

Note: For these changes to go into effect, end users should restart their clients.




Assigning User Permissions

To assign user permissions:

- Go go **Security > Permissions**. The Permission Assignment Table is displayed.
- For the permission you want to assign click **Edit** . The Permission Assignment Editor opens.
- Click the **Users** tab.
- From **Available Users**, select the user name and specify the permission:
 - To allow the permission for the user, click **Add to allowed list** . The user name is added to the Allowed Users list on the left.
 - To deny the permission for the user, click **Add to denied list** . The user name is added to the Denied Users list on the right.



Tips:

- To remove a user from the allowed or denied list, select the name and then click either **Remove from Allowed List**  or **Remove from Denied List** . The name is moved back to the Available Users list.
- If the user you want to assign is not listed, click **Add User** , type the user's name in the dialog, and then click OK. A name can contain any alphanumeric character except : ; / | or \. The new user is added to the Available Users list so you can assign a permission.

- Click **Continue** to save your changes and close the editor. The Permissions Assignment Table updates to show the newly assigned user permissions.

Note: For these changes to go into effect, end users should restart their clients.

Certificates

Managing SAML Certificates

Pipeline Pilot includes some support for user authentication based on the Security Assertion Markup Language (SAML) standard, when the client server communication is via Simple Object Access Protocol (SOAP).

The WS-Security standard defines how security information can be included in a SOAP header. When this takes the form of simple user names and passwords (Username Token profile), the credentials are validated against the authentication method configured for the Pipeline Pilot server.

In addition, the Pipeline Pilot SOAP handler service supports the passing of SAML-based authentication data (Binary Security Token profile) in the SOAP header. To enable this, you must import SAML certificates into a Trusted Certificate store for those servers from which Pipeline Pilot is to accept SAML assertions in SOAP-based requests.

Overview

Security Assertion Markup Language (SAML) is a standard set of open standards that allow web services to securely exchange user authentication and authorization data. The most important issue addressed by SAML is single sign on.

A SAML assertion is an electronic document that uses a digital signature to bind a public key with an identity – information such as the name of a person or an organization, address, etc. The certificate can then be used to verify that a public key belongs to an individual or organization. For a web trust scheme, the digital signature is either the user (self-signed certificate) or other users (endorsements). The signatures are attestations by the certificate signer that the ID information and public key belong together.

Web services can use one of many SAML token profiles that dictate the structure and role of SAML assertions within message headers sent to or from a service. A profile may use one of several confirmation methods to dictate how the identity of the a SAML assertion's subject is validated. Pipeline Pilot supports profiles using the Sender-Vouches subject confirmation method for inbound and outbound messages.

- **Inbound support:** The ability for an application to call a Pipeline Pilot web service using SAML Sender-Vouches Subject Confirmation to specify the caller's identity.
- **Outbound support:** The ability for the Pipeline Pilot SOAP Connector to access a SAML web service using a SAML Sender-Vouches Subject Confirmation with a self-signed certificate.

Use the SAML Certificates page to configure and manage SAML identity and trust relationships.

SAML Certificate Features

Pipeline Pilot uses digital certificates to sign SAML assertions when making outbound calls to remote services. The following certificate types are supported:

- CA-authorized certificates obtained within or outside of your organization
- Self-signed certificates generated in the Admin Portal

To work with SAML certificates in Pipeline Pilot, the following features are required:

- **Pipeline Pilot key pair:** The server must have a key pair, comprised of a public key embedded in a signed certificate and private key. These keys work together to establish an encrypted connection. A key pair with a self-signed certificate for Pipeline Pilot can be generated in the Admin Portal at the SSL Certificate page. You may also obtain a certificate signed by a certificate authority (CA) through a trusted vendor.

Note: The certificate must contain a subject (the identity of the certificate or web site owner). The Pipeline Pilot Server uses all of this information in the certificate to establish a secure SSL connection.

- **SAML key store:** This is where the private key pair used to sign SAML assertions is kept.
- **Trusted certificate store:** Contains the list of certificates of clients that are allowed to make web service calls to the Pipeline Pilot Server.

Importing Key Pairs into the SAML Stores

Requirement: To import a Pipeline Pilot key pair, the private key and certificate files need to be created at the [SSL Certificate page](#).

To import a key pair into the SAML stores:

1. Go to **Security > SAML Certificate**.
2. Click **Import SSL Key Pair**.

Two files are created and added to your server in "<install_root>/web/conf":

- truststore.jks
- keystore.jks

After the key pair is successfully imported, the SAML Certificate tab displays details about the subject and issuer information contained in your certificate.

| Subject Information | Issuer Information |
|---------------------|--------------------|
| C=US | C=US |
| ST=California | ST=California |
| L=Carlsbad | L=Carlsbad |
| O=MyCompany | O=MyCompany |
| OU=MyDepartment | OU=MyDepartment |

Re-Import SSL Key Pair

Note: If you make changes to the SSL certificate, you should update your SAML key pair by clicking **Re-Import SSL Key Pair**. However, when the server's SAML key changes, any remote services that already trust it will need to be updated with the new certificate.

Customizing SAML Issuer Tokens

You can customize the following settings in the SAML Certificate tab to control the behavior of outbound web service calls.

- **Issuer URI:** An arbitrary string used to uniquely identify this Pipeline Pilot Server with remote services. It does not need to be tied to the certificate's subject or issuer, however, it should uniquely identify the particular instance of Pipeline Pilot running on the host machine. Some combination of

hostname and port number is recommended.

- **Time to Live (seconds):** How many seconds an assertion generated by this Pipeline Pilot Server is valid. While valid, the assertion can be used as an authentication token by a remote service. If SAML assertions expire before they reach the remote service, increase the value.
- **Time to Live Offset (seconds):** Because the token interacts with services on other systems, the *Time to Live* setting must be offset slightly to account for differences in clocks. This setting will push back the validity window into the past, to accommodate remote services whose clocks are running behind. Without an offset, such services may reject assertions as being "Not Yet Valid". (Generally, the offset should be a fraction of the time to live.)

SAML Issuer Token Parameters

Issuer URI: MyServer

Time to Live (seconds): 10

Time to Live Offset (seconds): 3

Save Parameters View / Export SAML Metadata

Tip: If you change any parameters, click **Save Parameters** to save your new settings.

Sharing SAML Metadata Generated by Pipeline Pilot

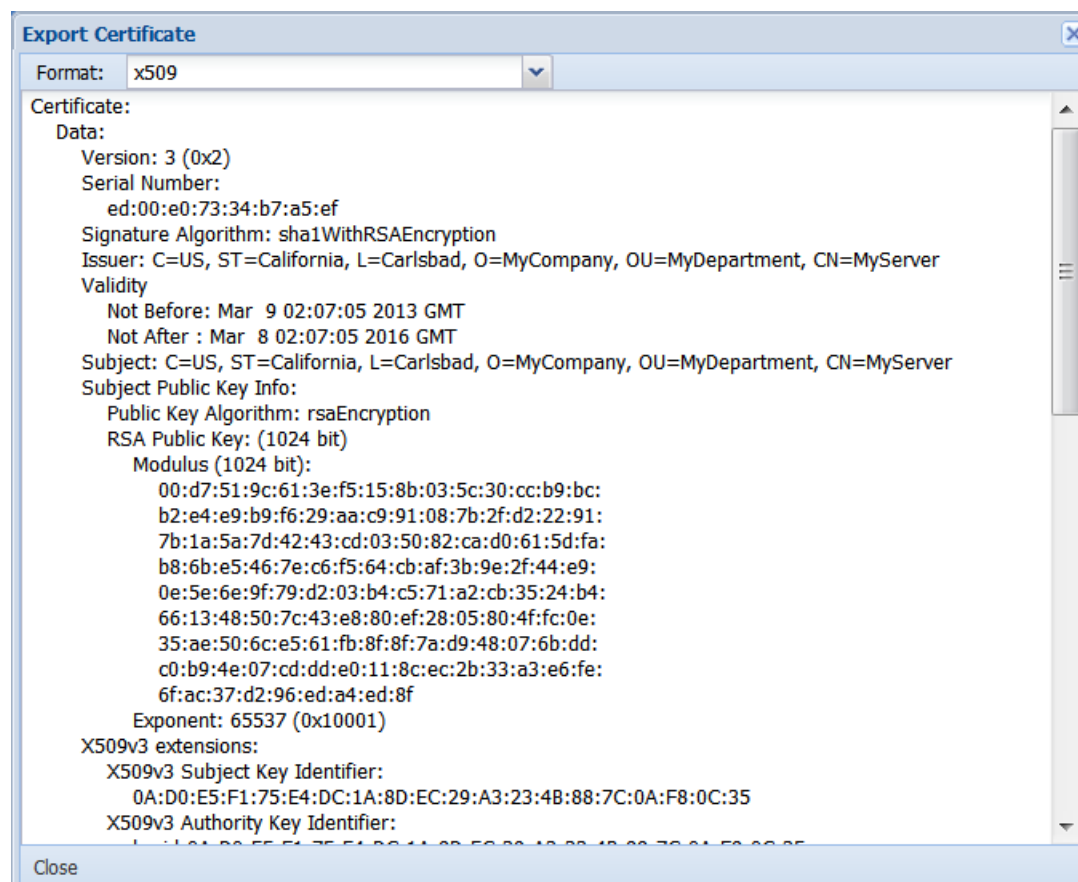
You can share Pipeline Pilot Server's SAML metadata with other servers to establish a trust relationship for making outbound calls from Pipeline Pilot. Several different metadata formats are available so that your SAML certificate can work with other service providers.

Supported metadata formats include:

- **PEM:** A minimal chunk of data containing only the certificate encoded for transport.
- **Pipeline Pilot SAML2 SP Metadata:** A format used for facilitating interoperability between Pipeline Pilot Servers.
- **Weblogic SAML2 IDP Metadata:** Format used by Weblogic web servers.
- **x509:** A format providing extended information.

To view SAML metadata:

1. Click **View/Export SAML Metadata**. An Export Certificate dialog opens.
2. Select a format from the dropdown. The dialog updates to display the metadata in the chosen format.
3. You can copy this information to share it with others in your organization.



Adding Trusted Certificates

To allow inbound web service calls to the Pipeline Pilot Server to authenticate using SAML assertions, you must add certificates of remote clients that you trust into your Pipeline Pilot Server's SAML trust store.

When an external service tries to access a protocol on the Pipeline Pilot service using a WS-Security protected SOAP message with a SAML assertion, the Pipeline Pilot Server will attempt to verify the integrity of the request. It will validate the digital signature, checking that the message was signed by an entity that owns a trusted certificate. If the signature is valid, the assertion issuer is matched against a list of permitted issuers for that certificate. If a match can be made, authorization is granted. If not, authorization is not granted and the request is rejected.

Requirement: To add a certificate into the trust store, you need to obtain it from the remote client.

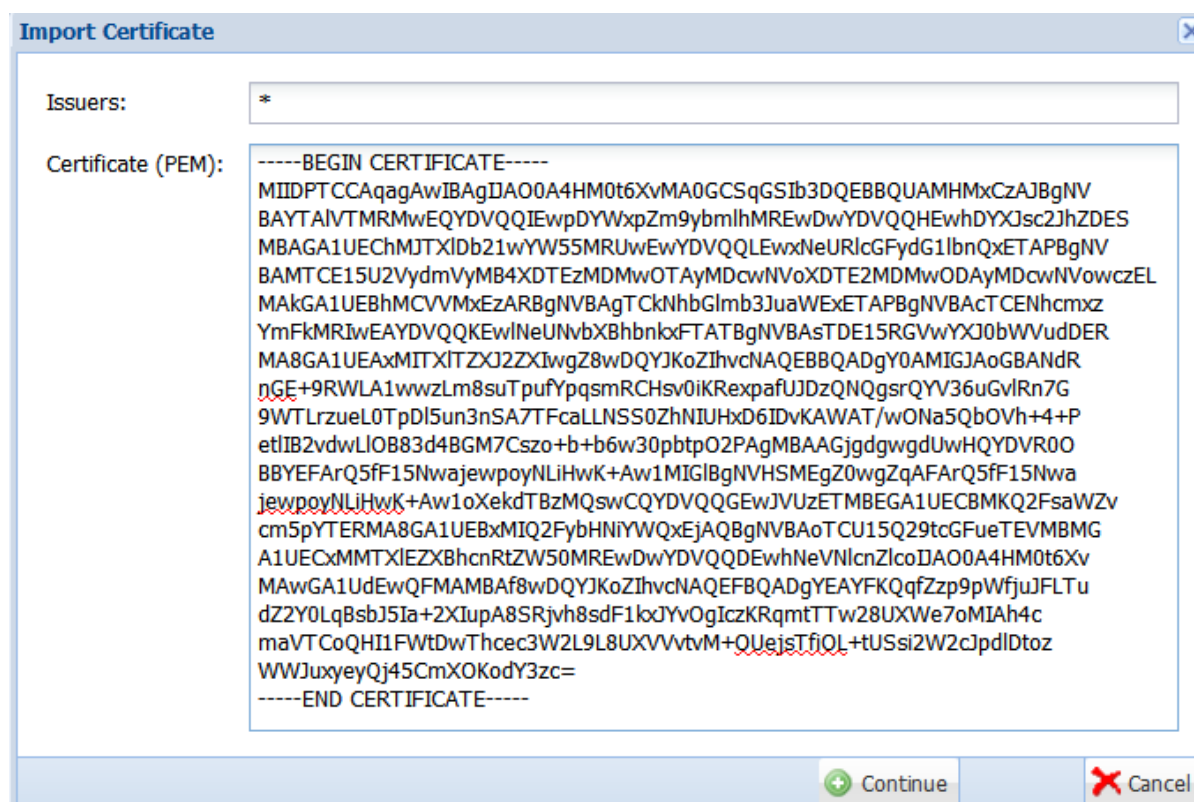
To add trusted certificates to the key store:

Obtain and open the certificate file containing a PEM block of data from the server to be trusted.

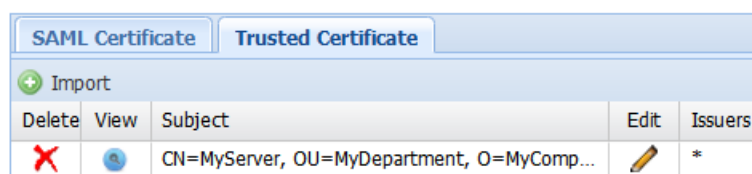
1. Go to **Security > SAML Certificate**.
2. Click the **Trusted Certificates** tab.
3. Click **Import**. An Import Certificate dialog opens where you can add the information.

To add trusted certificates to the key store:

1. Obtain and open the certificate file containing a PEM block of data from the server to be trusted.
2. In the Import Certificate dialog, add the following information:
 - **Issuers:** By default, this field contains an * character to mean that any server can present this certificate to Pipeline Pilot Server and will be allowed to use the service. You can further restrict service usage by adding the Issuer URI values for the servers that are allowed to use this certificate.
 - **Certificate:** Paste the information from your certificate file, from the text "-----BEGIN CERTIFICATE-----" to "-----END CERTIFICATE-----".



3. Click **Continue**. The certificate is saved to the server.



Updating a Trusted Certificate in the Trust Store

To restrict servers that can use a certificate after it is added to the trust store:

1. Click **Edit** .
2. In the Issuer Editor that opens, type the issuer URI of each server that should be able to launch protocols on this Pipeline Pilot Server. Add each name on a separate line. Do not use separator characters.
3. Click **Save**.

To remove a certificate from the trust store:

➤ Click **Remove** .

To view the certificate details (key pair information and certificate/web site owner identity):

➤ Click **View** .

Managing SSL Certificates

Secure Sockets Layer (SSL) is a standard cryptographic protocol for establishing a secure link between a server and a client (for example, a web server and a browser). SSL allows sensitive information such as login credentials or bank account data to be transmitted securely.

Pipeline Pilot supports SSL encryption of sensitive client server communications. The HTTPS protocol employs SSL to securely transmit HTTP traffic. The Pipeline Pilot Server exposes ports for both HTTP and HTTPS traffic.

For your production deployment of Pipeline Pilot, you must obtain and install a trusted SSL certificate from a recognized Certificate Authority to secure communication between the server and any client connecting through SSL.

SSL Certificates

The SSL protocol makes use of asymmetric cryptography to allow two parties to communicate securely over an encrypted channel. In such a scheme, the server owns a private key that is only known to the server. The server, however, distributes a public key to clients wishing to establish a secure connection. The client encrypts and transmits a message using the public key such that it can only be decrypted with the private key. Since only the server has the private key, the message cannot be intercepted and decrypted in transit.

The public key is encapsulated in an SSL certificate that also contains information about the server and expiration data and signatures of third parties that can vouch for the certificate's validity.

The client can use this additional information to decide whether to trust the certificate or not. Most browsers will only trust certificates signed by a handful of known authorities. If a certificate is signed by a different authority, it may be necessary to override the browser's trust mechanism.

Configuring SSL Certificates in the Admin Portal

There are various ways to acquire SSL certificates for use with Pipeline Pilot. The SSL Certificate page in the Administration Portal (Security > SSL Certificate) offers the following options:

- Obtain a certificate from a recognized authority (for example, generate a certificate by your company's internal authority).
- Generate your own self-signed certificate (that is, configure a certificate directly in the Admin Portal).
- Use the BIOVIA self-signed certificate (this is the default SSL setting that cannot be used as a SAML certificate).

SSL Certificate Guidelines

With SSL, a standard practice is to obtain a valid SSL certificate for your Pipeline Pilot installation from a recognized Certificate Authority (CA). An SSL certificate obtained from a CA verifies that a trusted third party has authenticated a certificate's validity. By trusting the CA, the browser can trust a certificate issued by the CA and confirm with end users that the site is secure.

An option for handling certificates on a production server is to have one generated by your company's internal CA. After a certificate file is generated along with the associated key pair file, you can import the pair into the Admin Portal (SSL Certificate page).

You can also purchase a certificate for your server host machine from a third-party CA. The cost depends on the certificate vendor and the level of trust you require. The more costly certificate products are

targeted at public web sites, where the signing authority makes efforts to establish valid ownership of a domain name. Less costly alternatives exist for intranet server use.

IMPORTANT! If you are deploying behind a load balancer, reverse proxy, or SSL terminator, be sure to obtain a certificate for the head node.

Using SSL Certificates from Recognized Authorities

Many signing authorities (CAs) are also available on the web for obtaining SSL certificates including:

- GoDaddy: <https://www.godaddy.com/>
- VeriSign: <https://www.verisign.com/>
- DigiCert: <https://www.digicert.com/>

Tip: When obtaining a certificate from a CA, indicate the applicable server software or server type. Select "Apache2" or "Apache2 with SSL".

After obtaining a certificate from a recognized authority, you can configure Pipeline Pilot to use it in the Admin Portal.

Support for Internal SSL Certificates

If you need to use internal SSL certificates with Pipeline Pilot, the following notes might serve as guidelines for this task:

| Certificate Type | Details |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Local Trust Per Computer/User | Users can decide if they want to trust a given certificate (current method for many users). Users have to actively trust the self-signed certificate versus ignoring or accepting the risk. |
| Local Trust through Group Policy | <ul style="list-style-type: none"> ■ Defined by IT Admin and published to computers. ■ Local IT administrators obtain the self-signed certificates and adds them to the acceptable certificates within their domain. ■ Practical if you do not have an Enterprise Certificate Authority (or do not want one) and do not want to purchase a certificate from the Internet. |
| Internet Trust | <ul style="list-style-type: none"> ■ When purchasing a public trusted certificate, the owner of the domain must approve the certificate, making it necessary to get a local IT department involved. ■ Internet Trusted works because there are standard "Trusted Root Certificate Authorities" that are common among Windows desktops as defined by Microsoft (by default). ■ The downside is that the certificates are expensive and tied to a given Fully Qualified Domain Name (FQDN, for example: <code>myinternalservername.biovia.net</code>). If you add servers, you need to purchase more certificates. |

| Certificate Type | Details |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Enterprise Trust | <p>Enterprise IT administrators can add additional CAs including their own. Allows an IT Department to create internal certificates that have the following features:</p> <ul style="list-style-type: none"> ■ Are bound to their domain created or managed by IT ■ Do not require additional payment ■ Are inherently trusted by all computers within their domain or other domains that also trust the Enterprise root certificate |

Note: For further details, see Windows Server Active Directory Certificate Services at [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831740\(v=ws.11\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831740(v=ws.11)).

Certificates and Certificate Chains

The Pipeline Pilot Server can accept either a basic server certificate or a certificate chain file which includes all intermediate signing authorities. The certificate or chain file must be formatted in PEM format (PKCS10) and consists of one or more "-----BEGIN CERTIFICATE-----" "-----END CERTIFICATE-----" sections ordered from leaf (your server's certificate) to root (the trusted root authority's certificate).

In previous versions of Pipeline Pilot, the chain file was separate from the server's certificate file. However, now the certificate file can either contain the simple server certificate or a full chain of certificates. If the certificate or chain is not in PEM format (recognizable by the "-----BEGIN CERTIFICATE-----" "-----END CERTIFICATE-----" sections), you will need to convert the file format to the appropriate type.

This can be done using the `openssl` command line executable installed on the Pipeline Pilot Server in the `apps/scitegic/openssl` package or by numerous online utilities.

Using Certificate Chains

SSL Certificates work on the concept of trust. All web browsers (Internet Explorer, Chrome, Firefox, Safari, etc.) are configured to trust the signatures of specific root authorities (for example, GoDaddy, DigiCert, VeriSign, etc.).

However, not all certificates are signed directly from these sites. For instance, a company might create its own intermediate certificate authority which is signed by the external root authority and this intermediate authority then creates server certificates for use within the company. When a browser connects to your server and inspects the SSL Certificate, the browser attempts to track back to a known root authority in order to establish trust.

For example, if server A which has a certificate signed by MyCompanyAuthority which in turn is signed by VeriSign. Chrome connects to server A's HTTPS port and discovers that it is signed by MyCompanyAuthority. Without additional information, Chrome does not know MyCompanyAuthority so cannot trust server A's certificate. This is where the certificate chain is employed. Instead of merely providing server A's certificate on the Pipeline Pilot Server, the chain provides server A's certificate along with the certificate information of MyCompanyAuthority and VeriSign establishing a trust chain. When Chrome connects to the HTTPS port, all three certificates are provided and Chrome is now able to track server A's certificate back to VeriSign which is a root authority trusted by Chrome.

Deciding Whether a Chain is Necessary

Whether a chain is necessary depends on your environment and how you obtained your certificate. In general, if you have a chain file available, try that first. In some cases, the file is provided by the authority split apart, in which case you will need to assemble the chain yourself. The multiple CERTIFICATE blocks should be ordered from server (leaf) to root authority.

Running the CSR Creation Process for Apache SSL

You can generate a certificate request file by running a Certificate Signing Request (CSR) for Apache SSL. This results in the following items:

- Private key file
- CSR file that you can send to the CA in support of your certificate purchase

To run the CSR creation process for Apache SSL:

1. Use the OpenSSL program, installed with the Pipeline Pilot server at: <install_root>/apps/scitegic/core/packages_<platform>/openssl.
2. Create a file 'subjectAltName_no_prompt.conf' with the following content

```
[ req ]
default_bits = 2048
distinguished_name = biovia_dn
x509_extensions = biovia_extensions
req_extensions = biovia_extensions
utf8 = yes
prompt = no
[ biovia_dn ]
countryName =
stateOrProvinceName =
localityName =
organizationName =
organizationalUnitName =
commonName =
emailAddress =
[ biovia_extensions ]
keyUsage = digitalSignature
extendedKeyUsage = serverAuth
basicConstraints = critical,CA:FALSE
subjectAltName = DNS:<server>
```

Complete the [biovia_dn] section with values for your organization. For subjectAltName, include the fully qualified domain name of your server and additional aliases if needed. For example: DNS:myserver.internal.mycompany.com, DNS:pipelinepilot.mycompany.com

3. Open a shell window to the openssl folder run a command in the form:

```
openssl req -new -newkey rsa -keyout ses-signed.key -out ses-signed.csr
-nodes -config "subjectAltName_no_prompt.conf"
```


where subjectAltName_no_prompt.conf is the path to the file you created in step 2.

Tips:

- Be sure to keep the generated key file `ses-signed.key` in a reliable location. You will need it when configuring the SSL certificate on the Pipeline Pilot Server.
- To generate a new server certificate, you will need to send the CSR file to the signing authority (CA).
- You can verify the contents of the CSR file by using the following command:
`openssl req -noout -text in ses-signed.csr`
- For further details on the `openssl` command, consult the online `openssl` documentation or your CA.

Configuring Pipeline Pilot to use a CA Certificate

To configure Pipeline Pilot to use a third-party certificate:

1. Copy the authority-signed certificate files to the folder "`<install_root>/web/conf`" and name the files as follows:
 - **Certificate file:** `ses-signed.crt`
 - **Private key file:** `ses-signed.key`
 - **Certificate chain file (optional):** `ses-signed.chain`
2. Log into the Admin Portal and go to **Security > SSL Certificate**.
3. Select option "1. Obtain a certificate from a recognized signing authority".
4. Click **Add Certificate Information** .
5. Paste all the information from the **Certificate File** and **Key File** into the online form.
6. For Firefox browser support, paste all of the information into the **Chain File** (explained below).
7. Click **Upload**.
8. Click **Save Option**.
9. Restart the server.

Using SSL Certificates with Firefox

For most browsers, the server publishing a certificate issued by a trusted authority is sufficient to eliminate security warnings when communicating via HTTPS. For some browsers, however, the server may need to publish an additional file that contains a chain of certificates used by the vendor to issue the server's certificate. Usually, certificate vendors will supply this data when issuing the certificate.

To install the chain file, paste the file contents, comprised of multiple certificates into the optional Chain File text pane of the SSL Certificate page in the Admin Portal.

Generating a Self-Signed Certificate

You can generate a self-signed certificate associated with your server host computer. Although this type of certificate is not generated by a third-party CA, it reduces the level of warnings. Modern browsers will indicate that this certificate is not from a certified signing authority, but users can confirm it as a trusted certificate in the browser's certificate management interface.

The Admin Portal includes a form for generating a self-signed certificate. Ensure that you enter the server name in a style that should be used by all web users - preferably the fully qualified domain name of the server.

To generate a self-signed certificate:

1. Log onto the Admin Portal and go to **Security > SSL Certificate**.
2. Select option "2. Generate your own self-signed certificate".
3. Use the online form to generate a self-signed certificate.

| | | |
|-------------------------------------------------------------------|--------------|--------------------------------------------------|
| Country Code (ISO 3166 two-letter country code) | US | Examples: US, CN, JP, IN, DE, RU, GB, BR, FR, IT |
| State or Province Name | California | |
| Locality or City Name | San Diego | |
| Organization or Company Name | MyCompany | |
| Organizational Unit or Department | MyDepartment | |
| Server Name (of the AEP server machine, as referenced by users) | MyServer | |
| <input type="button" value="Generate Certificate"/> | | |

4. After entering all the required information into the fields, click **Generate Certificate**.
5. Click **Save Option** (scroll up to access this option). The certificate will not be generated until you save.
6. Restart your server.

After you save the self-signed certificate in the SSL page the following files are created in "<install_root>/web/conf":

- ses-self-signed.crt
- ses-self-signed.key

Using the BIOVIA Self-Signed Certificate

Pipeline Pilot offers an out-of-the-box certificate that you can use with your server.

If you select this option, be aware of the following limitations:

- Since this certificate does not include the correct server name, modern browsers do not treat it as a trustworthy certificate.
- Users may encounter serious warning messages when this feature is implemented on your server.





To use a BIOVIA self-signed certificate:

1. In the Admin Portal, go to **Security > SSL Certificate**.
2. Select option "3. Use BIOVIA self-signed certificate (Default)".
3. Click **Save Option**.
4. Restart your server.

Tip: To view the information generated by the certificate, click .

Package Editors

The ability to control who can make changes to package resources is an important aspect of controlling and managing updates to the published services of Pipeline Pilot server. The Package Editors page allows you to assign editing rights to specific users or groups for specific packages not provided by BIOVIA. This page is similar to **Reports > Installed Packages**, but also indicates users and groups allowed to edit the packages.

| ID | Vendor | Name | Editors |
|------------------------|-------------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| scidev/admin | SciTegicDev | Admin regression tests conf | <div>  <div> <div>exampleadmin exampledev</div> <div>admin/dev/AdminUsers</div> </div>  </div> |
| scidev/adminconfigtest | SciTegicDev | Admin command line config regression test | <div>  <div>None</div>  </div> |

Click the gear icon to make changes. You can add one user name or group name per line, and then click the check mark button to confirm. Click the X button to close the update view without making any changes.

A user who has the right to edit a package can make changes to a packaged protocol or component in the Pipeline Pilot Client and save it back to the package. Other users are blocked from doing this.

Note: When a new package is created using the "pkgutil -n" command (see the Application Packaging guide), the current user is automatically assigned editing rights. This will generally be an administrator, who can set up other users or groups for editing the package, using the **Package Editors** page.

JAAS Configuration

There are some scenarios in which Java based components can use a plugable login module to authenticate against remote servers. For example, secured Hadoop clusters use Kerberos as their security framework. Consequently Java based components that want to connect to a secured Hadoop cluster typically use the following login module:

```
com.sun.security.auth.module.Krb5LoginModule.
```

A Java Authentication and Authorization Service (JAAS) configuration file is used to define one or more login modules for a Java class. The JAAS configuration file for Pipeline Pilot, `jaas.conf` is located in `<pps_install>/config/jaas/`. Note that this file is not created until you modify it using the **Security > JAAS Config** page in the Pipeline Pilot Admin Portal. Template configuration data is provided to support configurations with Cloudera and Progress/Data Direct Hive and Impala JDBC drivers.

For more information, see:

<http://docs.oracle.com/javase/8/docs/technotes/guides/security/jgss/tutorials/LoginConfigFile.html>.

Pipeline Pilot JDBC Connections

To support integration with Hadoop Hive, the following JDBC Driver class names automatically use the JAAS configuration entered via the **Security > JAAS Config** page in the Pipeline Pilot Admin Portal:

- `com.ddtek.jdbc.hive.HiveDriver`
- `org.apache.hive.jdbc.HiveDriver`
- `com.cloudera.hive.jdbc41.HS1Driver`
- `com.cloudera.hive.jdbc41.HS2Driver`

3D Passport Service Settings

3D Passport is Dassault Systèmes' user authentication server for the 3DS Platform. To use the service you must configure Cross-Origin Resource Sharing (CORS) and import certificates for the 3D Passport servers.

Additionally the Pipeline Pilot server must have a valid, signed certificate from a recognized certifying authority that can be added from **Security > SSL Certificate**.

To configure 3D Passport Settings

1. Open **Security > Passport Service**.
2. Set **Enable 3D Passport Service** to **Yes**.
3. Set **3D Dashboard Server for Cross-origin Requests** to the URL for the 3D Passport server.
4. Click **Save**.
5. Click **Add certificate** to import the signed 3D Passport Server certificate.

Chapter 5:

Setup and Configuration

Catalog Settings

The Administration Portal [Catalog Search](#) feature and the search feature in the Pipeline Pilot Client require a searchable catalog of protocols, components, and parameter names. Use the **Setup > Catalog Settings** page to control how the catalog is updated.

Required Permissions

The following permissions are required to change the catalog settings:

- *Platform\Administration\Logon* permissions
- Group membership to *Platform\Administrators*. When run by the scheduler, the user for the updater is *_PP_ServiceAccount*, which is always able to read the entire XMLDB.

Updating Catalog Index

You can build the catalog as needed by following these steps:

1. From the **Sync Catalog Index** section, choose whether to **Clear Catalog Before Sync**. Clearing the catalog first ensures a complete rebuild, but takes longer to build. Otherwise, the catalog will be updated based on the adds, deletes, and moves of items in the XMLDB.
2. Click **Sync Catalog Now**.
3. Review the Statistics from **Most Recent Sync Job**.

Scheduling Catalog Sync Settings

Use the Schedule Catalog Sync settings to update the catalog at regular intervals.

1. From the **Schedule Catalog Sync** section, choose whether to **Clear Catalog Before Sync**. Clearing the catalog first ensures a complete rebuild, but takes longer to build. Otherwise, the catalog will be updated based on the adds, deletes, and moves of items in the XMLDB.
2. Click **Save**.

Updating Sync Settings

Use the Sync Settings to control automatic catalog updates, the level of logging, and whether to include protocols, components, and parameters in the user folders.

1. From the **Sync Settings** section, edit the settings as required:
 - **Incremental Index:** Update the catalog whenever a user adds, deletes, or moves a protocol, component, or parameter.
 - **Sync Log Level:** Choose between OFF, INFO (summary), and DEBUG (verbose). Logging is reported in the Tomcat log files.
 - **Include User Folders:** Choose whether to include protocols, component, and parameter names saved in user folders.
2. Click **Save**.

3. Restart the Tomcat Server:
 - a. Open **Maintenance > Manage Server**.
 - b. Click **Restart Tomcat**.

Data Sources

Creating Data Source Connections

You can configure the following types of data source connections:

- **ODBC (PP)**: ODBC data sources using the BIOVIA-supplied drivers defined in the Admin Portal.
- **JDBC**: JDBC data sources defined in the Admin Portal.
- **ODBC (DSN)**: Data sources defined as ODBC DSNs on your server, which require administrator log into that server.
- **MongoDB**: Mongo data sources defined in the Admin Portal.

Note: You can also use the Admin Portal to export and import your data source configurations to share connections with other servers.

To add a new data source:

1. Go to **Setup > Data Sources**. The Data Sources page opens. All data sources that are currently available to use with your server are displayed in the Data Source List. This list is blank if no data sources are currently configured.

| Data Source List | Description | Type |
|--------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|-----------|
| Direct80_sequoia_rhlinux | Direct 8.0 instance on Sequoia, RedHat Linux | ODBC (PP) |
| GEMSTest | | JDBC |
| Mongo_test | MongoDB | MongoDB |
| pilotdb_dd | Scitegic 7.0 Oracle Wire Protocol | JDBC |
| <div> <input type="button" value="Add Data Source"/> <input type="button" value="Import Data Sources"/> </div> | | |
| <div> <input type="button" value="Export All Data Sources"/> <input type="button" value="Export Selected Data Source"/> </div> | | |

2. Click **Add Data Source**. The Add Data Source form opens on the right.
3. Enter a name and description for your new data source.
4. Select the data source type: **JDBC**, **ODBC (PP)**, **ODBC (DSN)**, or **MongoDB**.
5. Enter the required information for the type of data source (*see below for details*).
6. Click **Save**.

| Data Source List | Description | Type |
|--------------------------|----------------------------------------------|-----------|
| Direct80_sequoia_rhlinux | Direct 8.0 instance on Sequoia, RedHat Linux | ODBC (PP) |
| GEMSTest | | JDBC |
| Mongo_test | MongoDB | MongoDB |
| pilotdb_dd | Scitegic 7.0 Oracle Wire Protocol | JDBC |

Edit Data Source

Name: Direct80_sequoia_rhlinux

Description: Direct 8.0 instance on Sequoia, RedHat Linux

Type: ODBC (PP)

Access Privileges: +

Driver: Oracle

Driver Version: Latest

Server: srv-sr-sequoia.pn.mdli.com

Port: 1521

ServiceID: QAFRAML6

Service Name:

Connection Settings:


Connection Timeout:

Optional DB Username: directqa_80

Optional DB Password:

Advanced Settings: +

Tips:

- To view the configuration for an existing data source, select its name from the Data Source List. Details are displayed in the form on the right.
- To update the configuration, change the appropriate settings in the form, and then click **Save**.
- To delete the data source, click .

ODBC (PP) Data Sources

ODBC (PP) data sources can be used with SQL components.

To create an ODBC data source using an BIOVIA-provided database driver:

1. In the Add Data Source form, select the correct **Driver**.
2. Select "Latest" as the **Driver Version**. (Always use the latest driver installed with the platform).
3. Fill in the rest of the required information, including the **Server** and **Port**, and the **ServiceID**, **Service Name**, or **Database** depending on the driver selected.

Tip: The ODBC (PP) data sources are only intended for the Data Direct drivers that are supplied by BIOVIA and installed on the server. If you need a different type of ODBC driver, configure it as an **ODBC (DSN)** data source.

JDBC Data Sources

JDBC data sources can be used with SQL components.

To create a JDBC data source:

1. In the Add Data Source form, select the **Driver**.
2. Enter the **Connection String**.
If the driver is recognized (the [Squirrel](#) drivers file is used), a template connection string is provided when the driver is selected.
3. When using Oracle, it is possible to configure access to multiple servers or RAC using the tnsnames.ora syntax, for example:

```
jdbc:oracle:thin:@(DESCRIPTION= (ADDRESS=(PROTOCOL=TCP) (HOST=server1) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST=server2) (PORT=1521)) (CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=myn)))
```

| Data Source List | Description | Type |
|------------------|---------------------------------|------------|
| GEMSTest | | JDBC |
| pilotdb_dd | BIOVIA 7.1 Oracle Wire Protocol | ODBC (DSN) |
| PP Derby Example | | JDBC |

Edit Data Source

Name: GEMSTest

Description:

Type: JDBC

Access Privileges:

Driver: org.apache.derby.jdbc.ClientDriver40

Connection String: jdbc:derby://localhost:9947/GEMSTest
TEMPLATE: jdbc:derby://<server>[:<port>]/<databaseName>[:<URL attribute>=<value>]

Connection Timeout:

Optional DB Username: scitegicadmin

Optional DB Password: *****

JAAS Config File:

KRB5 Initialization File:

Advanced Settings:

Query Service Settings:

Note: To make a driver Jar file available for use, it must first be uploaded to the server. Click **Import JDBC Driver**, browse to the driver Jar file, and click **Upload**.

JAAS Config File

If the JDBC data source uses Kerberos or another Java authentication and authorization service pluggable module, copy the configuration file (for example, `myjaas.conf`) to `<pps_install>/config/jaas` and enter the file name in the **JAAS Config File** field. If you leave this field empty, the default JAAS configuration file in that folder is used.

KRB5 Initialization File

If you are using Kerberos authentication, copy the Kerberos initialization file (for example, `mykrb5.ini`) to `<pps_install>/config` and enter the file name in the **KRB5 Initialization File** field.

ODBC (DSN) Data Sources

ODBC (DSN) data sources can be used with SQL components.

To create an ODBC (DSN) data source:

1. Log into the server machine and create a DSN using the ODBC Administrator application.
2. In the Add Data Source form, select the **DSN** from the list of available server DSNs on that machine.

Tip: This should not be required. Pipeline Pilot automatically recognizes all System DSNs defined on the server and creates an ODBC (DSN) Data Source for them.

MongoDB Data Sources

To create an MongoDB data source:

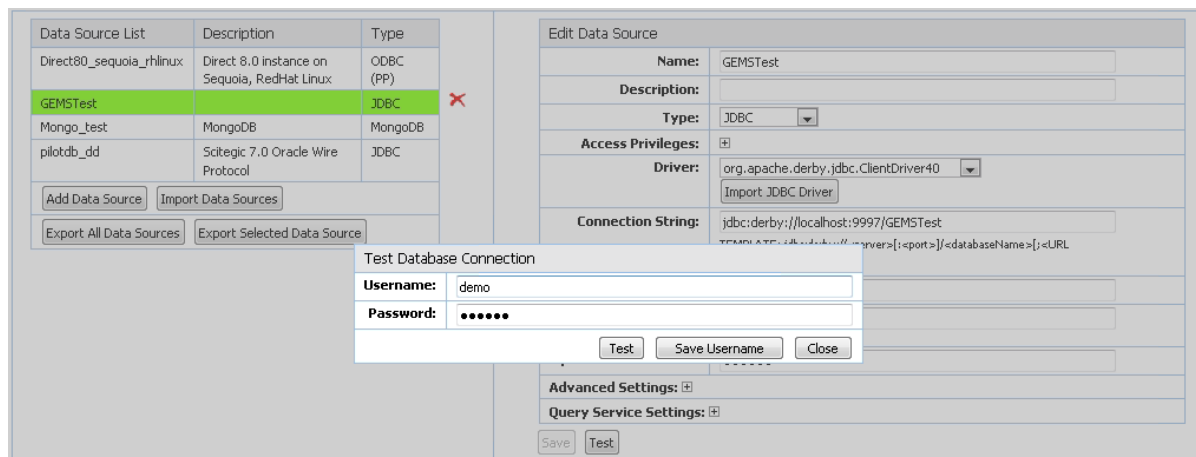
- Specify the **Server** and **Port**.

Testing Data Source Connections

After you configure a data source on your server, always perform a connection test to ensure that everything is working properly.

To test an SQL connection for the currently selected Data Source (even if you haven't saved any changes):

1. Click **Test**. A dialog is displayed.
2. If you have not already specified the optional username and password for the Data Source, specify that information.
3. Click **Test**.
4. If everything is correct, the message "Login Successful" is displayed. If there is a problem logging in, an error message is displayed that indicates the type of problem (for example, the wrong server name or incorrect username/password).



Advanced Data Source Connection Configurations

Depending on the type of database defined for a data source connection certain advanced settings can be configured (Setup > Data Sources).

Connection Pooling

For ODBC (PP) and JDBC data sources, you can specify a connection timeout. This setting determines how long a closed database connection will persist in a pooled server process.

To enable connection pooling for a data source:

- Specify the value of the **Connection Timeout** property in seconds (there is no pooling if the value is unset or "0").

| Edit Data Source | |
|------------------------------|-----------------------------------|
| Name: | pilotdb_dd |
| Description: | Scitegic 7.0 Oracle Wire Protocol |
| Type: | ODBC (PP) ▼ |
| Access Privileges: | + |
| Driver: | ▼ |
| Driver Version: | Latest ▼ |
| Server: | |
| Port: | |
| ServiceID: | |
| Connection Settings: | |
| Connection Timeout: | 60 |
| Optional DB Username: | |
| Optional DB Password: | |
| Advanced Settings: | + |

Notes:

- When a connection is closed in a pooled server process, it remains open for **Connection Timeout** seconds.
- If another protocol job in the same process tries to open the same database connection, it reuses the existing open connection, eliminating the overhead of making the connection.
- If the timeout is exceeded before a new request is made to the connection or the server process terminates, the connection is physically closed.

Security

Data source definitions are saved in an encrypted data file on the server. To enhance security, you can configure how users access your data sources.

To configure data source access rights:

1. In the Add or Edit Data Source form, expand **Access Privileges** and enter the required information.
2. To restrict data source access to a set of groups or users, select "Use Data Source" as the **Access Level**. This allows users to access the data source, but not view its definitions using the *List Data Sources* and *Manage Data Sources* components. Enter an **Entity Name** and click **Add/Edit**.
3. To restrict access to only members of the specified set of groups or users, select the "everybody" group and change the **Access Level** to "None" .
4. To allow users to bypass prompts for their usernames and passwords, specify an **Optional DB Username** and **Optional DB Password**. This eliminates the need for users to enter this information manually when they access the data source. By storing this information in one central and encrypted location (instead of in multiple protocols created by end users), your database access will

be more secure.

| Edit Data Source | | | | | | | | | | | | | | | | | |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|--------|------|--------|--------------|--------|--|-----------|-----------------|--|------|-------------|--------------|--------|----------------------|--------------------|
| Name: | <input type="text" value="pilotdb_dd"/> | | | | | | | | | | | | | | | | |
| Description: | <input type="text" value="Scitegic 7.0 Oracle Wire Protocol"/> | | | | | | | | | | | | | | | | |
| Type: | ODBC (PP) ▼ | | | | | | | | | | | | | | | | |
| Access Privileges: | <div> <input type="checkbox"/> <table border="1"> <thead> <tr> <th>Type</th> <th>Entity</th> <th>Access Level</th> <th>Remove</th> </tr> </thead> <tbody> <tr> <td></td> <td>everybody</td> <td>Use Data Source</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Type</th> <th>Entity Name</th> <th>Access Level</th> </tr> </thead> <tbody> <tr> <td>User ▼</td> <td><input type="text"/></td> <td>Edit Data Source ▼</td> </tr> </tbody> </table> <div> <input type="button" value="Add/Edit"/> <input type="button" value="Clear"/> </div> </div> | | | Type | Entity | Access Level | Remove | | everybody | Use Data Source | | Type | Entity Name | Access Level | User ▼ | <input type="text"/> | Edit Data Source ▼ |
| Type | Entity | Access Level | Remove | | | | | | | | | | | | | | |
| | everybody | Use Data Source | | | | | | | | | | | | | | | |
| Type | Entity Name | Access Level | | | | | | | | | | | | | | | |
| User ▼ | <input type="text"/> | Edit Data Source ▼ | | | | | | | | | | | | | | | |

To limit usage of a data source by component:

1. Select the data source which you wish to restrict.
2. Expand the **Advanced Settings** section of the Add or Edit Data Source form.
3. Enter the name of the **Required Registrant** for the only component(s) which can access this data source.

Only components with this *Registrant* will be able to connect to this data source.

| Edit Data Source | | | | | | | | | | | | | | | | | |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|--------|------|--------|--------------|--------|--|-----------|-----------------|--|------|-------------|--------------|--------|----------------------|--------------------|
| Name: | <input type="text" value="pilotdb_dd"/> | | | | | | | | | | | | | | | | |
| Description: | <input type="text" value="Scitegic 7.0 Oracle Wire Protocol"/> | | | | | | | | | | | | | | | | |
| Type: | ODBC (PP) ▼ | | | | | | | | | | | | | | | | |
| Access Privileges: | <div> <input type="checkbox"/> <table border="1"> <thead> <tr> <th>Type</th> <th>Entity</th> <th>Access Level</th> <th>Remove</th> </tr> </thead> <tbody> <tr> <td></td> <td>everybody</td> <td>Use Data Source</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Type</th> <th>Entity Name</th> <th>Access Level</th> </tr> </thead> <tbody> <tr> <td>User ▼</td> <td><input type="text"/></td> <td>Edit Data Source ▼</td> </tr> </tbody> </table> <div> <input type="button" value="Add/Edit"/> <input type="button" value="Clear"/> </div> </div> | | | Type | Entity | Access Level | Remove | | everybody | Use Data Source | | Type | Entity Name | Access Level | User ▼ | <input type="text"/> | Edit Data Source ▼ |
| Type | Entity | Access Level | Remove | | | | | | | | | | | | | | |
| | everybody | Use Data Source | | | | | | | | | | | | | | | |
| Type | Entity Name | Access Level | | | | | | | | | | | | | | | |
| User ▼ | <input type="text"/> | Edit Data Source ▼ | | | | | | | | | | | | | | | |
| Driver: | <input type="text"/> | | | | | | | | | | | | | | | | |
| Driver Version: | Latest ▼ | | | | | | | | | | | | | | | | |
| Server: | <input type="text"/> | | | | | | | | | | | | | | | | |
| Port: | <input type="text"/> | | | | | | | | | | | | | | | | |
| ServiceID: | <input type="text"/> | | | | | | | | | | | | | | | | |
| Connection Settings: | <input type="text"/> | | | | | | | | | | | | | | | | |
| Connection Timeout: | <input type="text" value="60"/> | | | | | | | | | | | | | | | | |
| Optional DB Username: | <input type="text"/> | | | | | | | | | | | | | | | | |
| Optional DB Password: | <input type="text"/> | | | | | | | | | | | | | | | | |
| Advanced Settings: | <input type="checkbox"/> | | | | | | | | | | | | | | | | |
| Require PP Credentials: | <input type="checkbox"/> Require Pipeline Pilot Credentials | | | | | | | | | | | | | | | | |
| Initial SQL: | <input type="text"/> | | | | | | | | | | | | | | | | |
| Required Registrant: | <input type="text"/> | | | | | | | | | | | | | | | | |
| Application Info: | <input type="text"/> | | | | | | | | | | | | | | | | |

Tip: To identify its *Owner* (Registrant) in Pipeline Pilot, right-click a component and select **Show Versions**.

Authentication

More strict user identification and authentication for connecting to the data source can be configured to enhance security. These options are not available for MongoDB data sources.

To configure the data source identification method:

1. Expand the **Advanced Settings** section of the Add or Edit Data Source form.
2. To force all connections to the current data source to employ the username and password of the current Pipeline Pilot Server user for identification, check **Require PP Credentials**.
3. For JDBC data sources, the **Optional DB Username** and **Optional DB Password** can be used to connect to the database. To use these credentials, check **Proxy Authentication**.

Note: If **Require PP Credentials** and **Proxy Authentication** are both selected for a JDBC data source, the specified username and password are used for the connection but the Pipeline Pilot Server user credentials are used for the proxy authentication.

Initializing the Connection

If a data source requires that SQL statements are executed every time a connection is made, this can be configured. This option is not available for MongoDB data sources.

To initialize the connection:

- Enter a single SQL statement in **Initial SQL**. Do *not* add a semicolon at the end of the SQL statement.
- If you are specifying multiple statements, ensure that you are using the appropriate database conventions. In Oracle, to modify multiple data sources' settings for a session, wrap them in an anonymous PL/SQL block:

```
BEGIN
    execute immediate ('ALTER SESSION SET NLS_COMP=LINGUISTIC');
    execute immediate ('ALTER SESSION SET NLS_SORT=BINARY_CI');
END;
```

- Use the **Initial SQL** parameter on a *SQL Open Connection* component to specify further SQL for individual connection instances.

Additional Information

If a data source requires additional information for its data sources, this can be configured. This option is not available for MongoDB data sources.

- Enter the required SQL statements in **Application Info**.

Query Service Settings

If a JDBC data source is used with a query service, the user for handling list tables can be customized and additional relationships between property names and values can be specified.

To configure query service settings:

1. Specify a **List DB Username** and **List DB Password** to user to create temporary tables for list handling. If no custom username and password are specified, the table will be created under connection user.
2. Enter any custom relationships between property names and values in **Query Service Properties** in the format:
'property_name1 = value1'; 'property_name2=value2'; ...

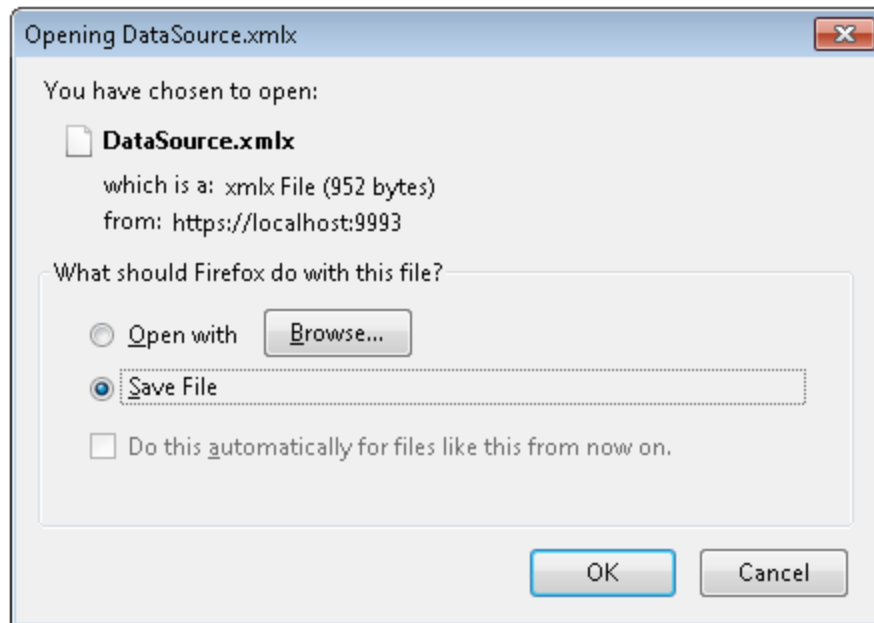
Exporting and Importing Data Sources

To facilitate sharing data source definitions between servers, you can export and import data sources.

To export data sources:

1. Go to **Setup > Data Sources**. The Data Sources page opens.
2. To export all defined data sources on your server, click **Export All Data Sources**.
3. To export a specific data source, select it from the list of data sources and then click **Export Selected Data Source**.

| Data Source List | Description | Type |
|------------------------------------------------------------------|----------------------------------------------|-----------|
| Direct80_sequoia_rhlinux | Direct 8.0 instance on Sequoia, RedHat Linux | ODBC (PP) |
| GEMSTest | | JDBC |
| Mongo_test | MongoDB | MongoDB |
| pilotdb_dd | Scitegic 7.0 Oracle Wire Protocol | JDBC |
| <div> Add Data Source Import Data Sources </div> | | |
| <div> Export All Data Sources Export Selected Data Source </div> | | |



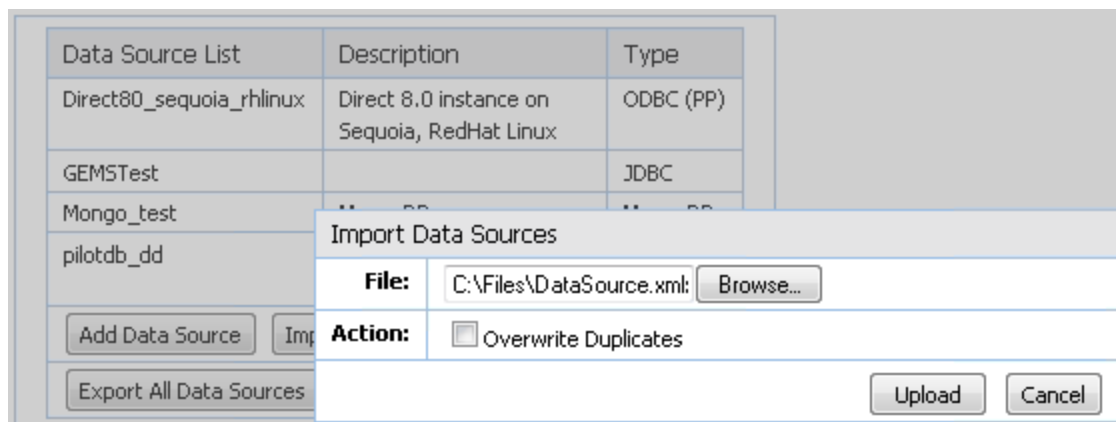
4. You are prompted to save an encrypted file ("DataSource.xmlx" or "DataSource-Data Source Name.xmlx" - for an individual data source) to the "Downloads" folder on your server (or wherever you want to save the file). This encrypted file contains your newly exported data source definition(s).

Note: You and/or other system administrators can then log into the Admin Portal on another server to import the data source definition file for use on this other server.

To import a data source:

1. Log into the Admin Portal on the server where you want to import the data sources.
2. Go to **Setup > Data Sources**. The Data Sources page opens.
3. Click **Import Data Sources**. A dialog opens.
4. For the **File** setting, browse to the location where "DataSource.xmlx" is available.

- To overwrite the content of existing data sources with duplicate names when importing, select **Overwrite Duplicates**.
- Click **Upload**. A brief report is displayed showing how many new data sources were imported and how many duplicates were loaded or ignored.



Tip: For further information, see the *Database Integration Guide* (Help Center > Developers tab > Server-side Integration).

Tagged Resources

A Tagged Resource is an entity that collects the information necessary to access a specific source of data, and gives it a name or “tag”, which can then be used by a Pipeline Pilot component as a key to look up this information when it needs to access the data. Specific information is required to access source of data. For example:

- Identity and/or location of the data
- Credentials to gain access
- Information that narrows down the specific data items you are looking for in a read operation

The details of the information needed depends on the category of data source, as should be clear from the following example data sources:

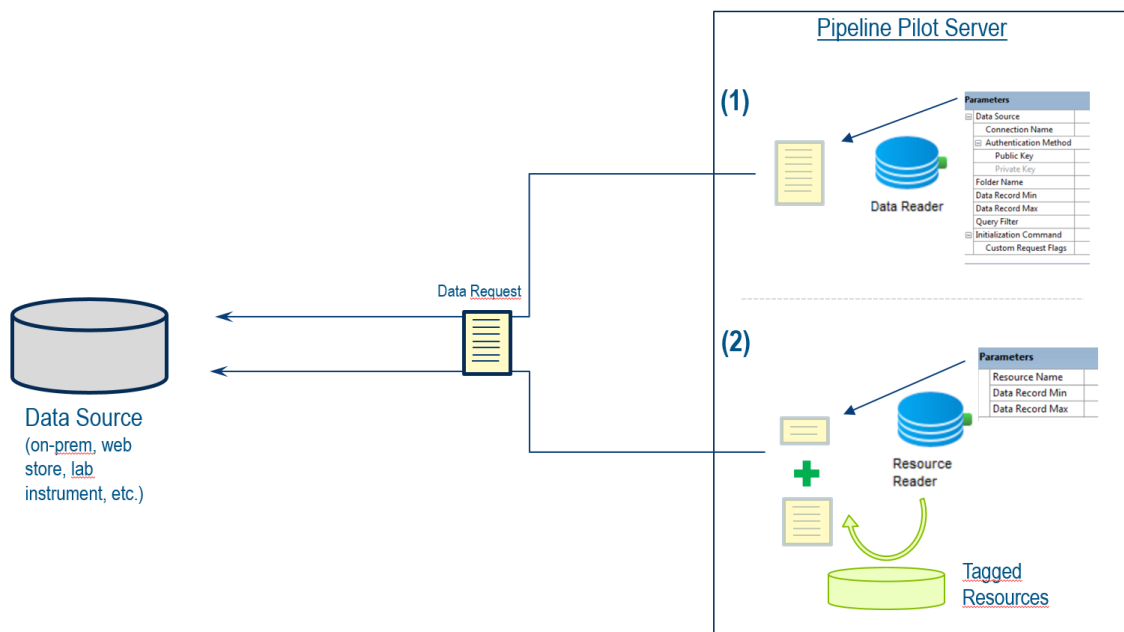
- RDMS database
- Online data store like Amazon S3 or SharePoint Online
- Networked laboratory instrument
- Pipeline Pilot cache

Note: Pipeline Pilot provides templates for creating a tagged resource for an external resource. For more information, see [Adding a Tagged Resource for an External Resource](#) on page 115.

The diagram below illustrates:

- A data access component parameterized with all the information required to access the target data
- A resource-oriented component, which is parameterized with the name of a Tagged Resource, from which it retrieves the pre-configured information associated with that resource. This information is combined with the values from a much smaller set of component parameters,

In both cases, the same request is sent to the data source, but constructed in different ways.



This approach has several advantages for Pipeline Pilot component developers and for protocol authors.

- **Simplified component interface:** A single parameter to hold the name of the Tagged Resource replaces the need for multiple parameters to hold all the information that the resource name represents. A protocol author needs to know only the name of the Tagged Resource, and not all the connection details – see diagram above.
- **Always use the latest information:** A by-product of the interface simplification (and abstraction) is that, over time, information may be added or changed in the resource definition, without any change required to components saved in existing protocols. For example, the resource password may require a periodic update; in the Tagged Resource scenario, this update only needs to be reflected in the stored resource definition, and does not require modification of every protocol that includes the data access component.
- **More secure credential storage:** If the resource information includes a password or other access token, this information is not persisted in any protocol which contains a resource access component. Instead, it is stored only within the encrypted Tagged Resource definition.
- **Administrative oversight:** A Tagged Resource is created by a Pipeline Pilot administrator, based upon a template provided by a component author. In this way, the administrator controls which resources are made available to protocol authors and users, and which data within those resources.
- **Access Control:** You can control access to the tagged resource by user and group. You can also limit use to a specific protocol package. For more information, see [User Access](#) on page 114.
- **Flexible Resource Definition:** In the design of a Tagged Resource, you can choose how specific to make the definition. It could include properties that restrict the resource to represent a very specific data query or with a specific limitation on data size, for example. Or it could be a much looser resource definition, which identifies the overall data source location, but does not try to define which data can be accessed or even which user name should be used. So, there is a range of choice between a more fully defined resource with a simple component interface and the other end of the spectrum where the resource is only broadly defined and the component is parameterized to

provide the other information - making it a more general purpose component at the cost of a more extensive parameter interface.

This model of encapsulated resource access is well established in Pipeline Pilot with the Data Source concept, but Data Source encapsulation is usable only for access to SQL databases (ODBC, JDBC) and MongoDB. The Tagged Resource concept generalizes this approach to any resource, such as the ones listed above.

Package Developer tasks

1. Create the templates in a ResourceTemplates.xml file in the XML Objects folder. See [Creating Tagged Resource Templates](#).
2. Create data access components that include a parameter for the Tagged Resource name. See [Creating data access components](#).

Creating Tagged Resource Templates

To use Tagged Resources, a package developer will need to develop and deploy XML-based templates that can be managed by an administrator.

Create a ResourceTemplates.xml file in the XML Objects folder of the package. ResourceTemplates.xml should have required properties plus additional properties for information that your components will require to access the data including locations, identifiers, credentials, filters, time restrictions, etc.

A template includes some standard properties. The first two relate to the template itself.

| | |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ResourceTemplate | The identifier for the template which effectively indicates the category of data source this template is used for. E.g., S3 Data Reader, Accuris Balance, Cache Reader |
| TemplateNotes | Any notes supplied in the template will appear in the Tagged Resource creation interface to guide the administrator when defining a Tagged Resource from this template. |
| Name | These properties should be in the template, but are left empty, to be filled out by the Administrator when defining a new Tagged Resource. These properties will be filled out as the name and description of the specific resource. E.g. for the example templates above, names might be Assay X2 Bucket, Balance Sc5, Project Cache Bio201706 |
| Description | |
| RequiredRegistrant | An optional package identifier. It can be pre-set in the template, but is also editable by the Administrator when defining or editing Tagged Resource. When defined for a resource, only components of that package can be used to query the properties of the Tagged Resource. |

The remainder of the properties in a Tagged Resource template represent whatever information a component needs to access the target data. For example, to access an Amazon S3 bucket, the template might include the AWS access key, the secret access key, the name of the bucket, and perhaps the path to a folder within the bucket.

The sample template file below is for the *Cache Resource Reader* component. Using this template, an Administrator can create multiple Tagged Resources, each for a specific Pipeline Pilot cache (which is itself password protected). A protocol author, using the *Cache Resource Reader*, references the Tagged Resource name, rather than the underlying cache identifier, which is managed as a Tagged Resource property.

```
<?xml version="1.0" encoding="UTF-8"?>
<sci:datadoc xmlns:sci="http://www.SciTegic.com/">
  <sci:data object="SciTegic.PropertyList.1">
    <sci:proplist>
      <sci:propval name="ResourceTemplate">Cache Data
Source</sci:propval>
      <sci:propval name="TemplateNotes">Use this template to create a
tagged resource based on a public, password-protected cache. Provide a name,
password, and access rights. A protocol author can use the Cache Resource
Reader component to access resource content. When the protocol is run, the
user is checked against the resource's access rights. Leave the CustomPath
blank unless the cache has been moved from the standard location for public
caches.</sci:propval>
      <sci:propval name="Name"/>
      <sci:propval name="Description"/>
      <sci:propval
name="RequiredRegistrant">scitegic/generic</sci:propval>
      <sci:propval name="CacheID"/>
      <sci:propval name="Password"/>
      <sci:propval name="CustomPath"/>
    </sci:proplist>
  </sci:data>
</sci:datadoc>
```

Note: A template file can contain multiple template definitions. In this case, the <sci:data> element is repeated as often as needed.

Creating data access components

Create data access components using any one of the standard component SDKs, Python, Java, etc. Each component should include a parameter for the Tagged Resource name. In your component script/code, use this value to extract all the properties defined for that resource. These will be the same properties that you defined in the template, but now they should be filled in with the values assigned by the Administrator for the specific resource named in the parameters for this component.

To query the properties belonging to a Tagged Resource, use the `findResourceByName()` method on the context object. This returns a property list, which contains the property values configured to allow your component to access the data source. The following example uses a *Cache Resource Reader* component that uses a Tagged Resource:



| Parameters | |
|---------------------|-------------------|
| Cache Resource Name | Assay Data 170527 |
| RangeMinimum | 1 |
| RangeMaximum | |

Parameters Runtime Implementation Information

Administrator Tasks

As a Pipeline Pilot administrator, you can define specific Tagged Resources, each based upon a template from an installed package. A Tagged Resource refers to a source of data accessible to the Pipeline Pilot installation, and by defining a Tagged Resource, you can control which data is to be accessed and by which users.

To create a new resource:

1. In the Administration Portal, open **Setup > Tagged Resources**.
2. Choose a Tagged Resource template from the list.
3. Click **New Resource**.
4. Configure the new Tagged Resource with the information necessary to access the target data, according to the properties in the form. This will include details to locate and identify the data and maybe credentials and other filters or restrictions.
5. Define the access rights for the resource, if you want to limit which users or groups can make use of this resource. For details about Access Rights, see [User Access](#).
6. Choose a unique name of the resource, which is how it is identified by protocol authors.
7. Press **Create**.
8. Publicize the name of the resource to protocol authors to use in their protocols.

Note: This process is analogous to what you may have already done in setting up a Data Source in the Administration Portal.

User Access

The Access Rights and Package Requirements fields allow you to control access to a tagged resource.

Access Rights Field

You can set access rights for groups and users. By default, tagged resources have access rights defined for "everybody". You can optionally define access for specific groups and users. Resource access can be set to Allow Access, Restrict Access, or Deny Access. Deny Access takes precedence over Allow Access. Restrict Access does not provide access, but does not block a subset group or user with the Allow Access setting.

To define access for groups and users:

1. Click the Access Rights table to open the Edit Access Rights dialog.
2. Make changes as needed:
 - Click an existing row and make changes to **Type**, **Entity**, or **Access** as needed. Click the red **X** to delete a row.
 - Click **Add Row** and define access for a specific group or user.
3. Click **OK**.

Package Requirement Field

Enter a package name to restrict the tagged resource to protocols that belong to that package. Leave this field blank to allow access to all protocols regardless of the package they belong to.

Adding a Tagged Resource for an External Resource

By creating a tagged resource for an external resource, Pipeline Pilot users can easily work with an external resource in a protocol without providing the URL or credentials every time they want to access it. Creating tagged resources also allows you to manage the resources centrally. Once you have set up a tagged resource, a protocol developer can reference the tagged resource as a resource connection. Pipeline Pilot provides templates for setting up tagged resources for a SharePoint Online Site and AWS S3 Site.

For information about using the tagged resources in a protocol, see *Users > Pipeline Pilot > Files and Data > Using a Resource Connection in your Protocol*.

Adding a Tagged Resource for SharePoint Online Site

1. Set up your SharePoint Site. See the SharePoint Online Help for details.

Open `<office365InstanceName>.sharepoint.com` as an administrator and do the following:

- From the Azure Active Directory register your Pipeline Pilot server as a new application (for example, "Pipeline Pilot on <my server>".)
- For **Redirect URI**, enter `<pipeline pilot server url>/oauth2/authorize`.
- After you register, record the client ID from the **Application (client) ID** field.
- Add API permissions for **AllSites.Manage**.
- Add a **Client Secret**.

2. In the Pipeline Pilot Administration Portal, open **Setup > Tagged Resources**.

3. Choose **SharePoint Online Site** from the list near the bottom of the page.

4. Click **New Resource**.

5. Complete the **Resource Details**:

- **Name and Description**: Enter a unique name for your resource and optional description.
- **ResourceURL**: Enter the resource URL using the following form:
`https://<office365InstanceName>.sharepoint.com`.
- **AuthorizationAuthorityURL**: Enter the resource URL using the following form:
`https://login.microsoftonline.com/<office365InstanceName>.onmicrosoft.com`.
- **ClientId**: Enter the client ID.
- **ClientSecret_password**: Client secret password.
- **Access Rights**: Access rights are granted to everybody by default. To change the access rights to this specific tagged resource, click **Access Rights** and make changes as needed. For details about Access Rights, see [User Access](#).

6. Click **Create**.

Adding a Tagged Resource for AWS S3 Site

1. Set up your AWS S3 instance and gather the following information:

- Access key ID
- Secret access key

See the Amazon AWS documentation for more information.

2. Log in to the AWS S3 site and do the following:
 - Create a bucket with a unique name.
 - Ensure that Block all public access is not selected on the Set Permissions screen.
3. From the Pipeline Pilot Administration Portal, open **Setup > Tagged Resources**.
4. Choose **AWS S3 Site** from the list near the bottom of the page.
5. Click **New Resource**.
6. Complete the **Resource Details**:
 - **Name and Description**: Enter a unique name for your resource and optional description.
 - **ResourceURL**: Enter the resource URL using the following form: <bucket name> . s3 . amazonaws . com.
 - **AWSAccessKeyId**: Enter the access key ID.
 - **AWSSecretKey**: Enter the secret key.
 - **Access Rights**: Access rights are granted to everybody by default. To change the access rights to this specific tagged resource, click **Access Rights** and make changes as needed. For details about Access Rights, see [User Access](#).
7. Click **Create**.

Protocol User Access to External Resources

For a user to run a protocol that references an external resource, the user must have access rights to the external resource, which is controlled by the Access Rights field in the tagged resource. If a user attempts to run a protocol without access rights, the protocol will fail with a “Resource not found” error. For details about Access Rights, see [User Access](#).

Additionally, Pipeline Pilot must be authorized to access the resource on the user’s behalf. If a user attempts to run a protocol without authorizing Pipeline Pilot to access the resource, the protocol will fail with a “Resource <resource_name> not authorized for user” error.

- For AWS S3, a single service account is used for access so no additional authorization is needed by individual users.
- For SharePoint Online, OAUTH2 authorization is used and requires each user to authorize Pipeline Pilot to access SharePoint Online on their behalf. Users can authorize Pipeline Pilot to access SharePoint Online from the Resource Authorization page available from the Pipeline Pilot Server Home Page. For more see “Authorizing the Resource” in Using a Resource Connection in your Protocol.

File Browser Access Settings

The files that can be browsed to from component parameters and web client applications can be customized. You can add shortcuts to directories containing commonly used data files and you can choose which folders of the server can be browsed to and modified.

File Browser Shortcuts


You can set up the server to include shortcuts to data directories where Pipeline Pilot component collections are installed. When client users browse for files on a Pipeline Pilot Server, shortcuts for these data directories are listed for quick access.

To add a shortcut:

1. Go to **Setup > File Browsing**.
2. Enter text in **Shortcut Name** that identifies the browser shortcut.
3. Enter the shortcut location in **Path**.
4. Click **Add**.

| Shortcut Name | Path |
|---------------|--------------------------------------------|
| Test Data | \\MyCompany.net\shares\thirdparty\dev\data |

To remove a shortcut:

- To remove an existing shortcut, click **Remove** .

File Browser Access

You can configure your server to manage access from connected Pipeline Pilot Clients and web clients to directories on the server. Files that can be used as input can be restricted and file modification (deleting, moving, renaming) can be controlled.

To control access:

1. Go to **Setup > Server Configuration**.
2. Configure the **File Browsing** setting:
 - To allow files anywhere on the server to be accessed, select **Unrestricted**.
 - To limit user access to the contents of the "scitegic/public" directory, select **Restricted**.
 - To limit user access to the contents of each "scitegic/public/users/[username]" directory, select **User Folder Only**.
3. Click **Save**.

To control modification:

1. Go to **Setup > Server Configuration**. The Server Configuration page opens.
2. Configure the **File Editing** setting:
 - To allow files anywhere on the server to be modified, select **Unrestricted**.
 - To limit user access to the contents of each "scitegic/public/users" directory, select **Restricted**.
3. Click **Save**.

Folder Locations

Managing Folder Locations

Overview

Pipeline Pilot maintains folders that perform specific functions on the server. These folders are assigned to default directories during installation and include the following:

- **User:** All users are given folders on the server for writing files used for long-term storage. The User directory holds these public user folders. If best practices at your organization require storing user folders on a separate partition or shared drive (e.g., to facilitate backups), you can redirect these folders to an alternate directory.
- **Jobs:** Stores temporary files created by protocol jobs. Besides acting as the current directory for job execution, it also holds temporary files while the job is running and job result files for the lifetime of a

job. Since some intermediate job and result files are large in size, you may need to redirect this folder to a different location outside of the software installation area.

- **Shared Public:** Stores files that need to be accessible to any file system that is shared across multiple servers. It enables shared files to be accessed when the server is running on a load balanced configuration.
- **Local Temp:** Use this folder to locate temporary job files to a file system that is local to the running job. This is useful when the server folder is located on a shared resource such as an NFS mount. Using a local temporary storage will improve protocol performance and reduce network loads on clusters and grids.
- **Upload:** Stores files uploaded to the server. This is an optional directory that administrators can configure if necessary. If an Upload directory is not defined (and the setting is left blank in the Admin Portal), User folders are used for this purpose.
- **XMLDB:** Stores protocols and components specific to the server. Some of these are public (Components and Protocols subfolders), and some are owned by specific users (Users subfolders). Besides a backup strategy for this data, you can relocate the directory to a more appropriate location for these valuable resources.

Example Folder Locations

| | | |
|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------|------------------------------|
| User Directory | <input type="text" value="\\server\share\folder\user"/> | Show Default |
| Details and Instructions | | |
| Jobs Directory | <input type="text" value="\\server\share\folder\jobs"/> | Show Default |
| Details and Instructions | | |
| Shared Public Directory | <input type="text" value="\\server\share\folder\public"/> | Show Default |
| Details and Instructions | | |
| Local Temp Directory | <input type="text" value="Same as job folder"/> | Show Default |
| Details and Instructions | | |
| Upload Directory | <input type="text"/> | Show Default |
| Details and Instructions | | |
| XMLDB Directory | <input type="text" value="\\server\share\folder\xmlpdb"/> | Show Default |
| Details and Instructions (Be sure to read this information before moving an XMLDB.) | | |

Redirecting Folders

Default directories are defined when Pipeline Pilot is installed on your server. You can redirect these folders at any time after installation. Depending on which folder you are redirecting, you will need to perform specific steps to ensure that folder features work as expected.

Redirecting the User Folder

All users are given folders on the server for writing files used for long-term storage. The User directory holds these public user folders. If best practices at your organization require storing user folders on a separate partition or shared drive (e.g., to facilitate backups), you can redirect these folders to an alternate directory.

IMPORTANT! Before you redirect a folder, confirm that no jobs are running on the same server (**Status > Running Jobs**).

To redirect public user folders:

1. Go to **Setup > Folder Locations**.
2. In **User Directory**, change the path so it points to the alternate directory.

Tips:

- To view the default path, click **Show Default**.
- To verify that the alternate directory is valid, click **Validate Locations**.

3. To save the new directory setting, click **Save**.
4. To enable the new directory on the server:
 - a. Immediately stop Pipeline Pilot.
 - b. Keep existing User folders and files by copying subfolder contents from the previous location to the new directory.
Default User directory location:
`<installation-path>/public/users`
 - c. Restart the server.

Redirecting the Jobs Folder

The Jobs folder is a directory that stores temporary files created by protocol jobs. Besides acting as the current directory for job execution, it also holds temporary files while the job is running and job result files for the lifetime of a job. Since some intermediate job and result files are large in size, you may need to redirect this folder to a different location outside of the software installation area.

IMPORTANT! Before you redirect a folder, confirm that no jobs are running on the same server (**Status > Running Jobs**).

To redirect the Jobs folder:

1. Go to **Setup > Folder Locations**.
2. In **Jobs Directory**, change the path so it points to the alternate directory.

Tips:

- To view the default path, click **Show Default**.
- To verify that the alternate directory is valid, click **Validate Locations**.

3. To save the new directory setting, click **Save**.
4. To enable the new directory on the server:
 - a. Immediately stop Pipeline Pilot.
 - b. Keep existing User folders and files by copying subfolder contents from the previous location to the new directory.

Default User directory location:
<installation-path>/web/job

- c. Restart the server.

Do not share a Jobs folder between different server installations. Each installation should have its own unique Jobs directory to prevent version problems.

Redirecting the Shared Public Folder

The Shared Public folder stores files that need to be accessible to any file system that is shared across multiple servers. It enables shared files to be accessed when the server is running on a load balanced configuration.

IMPORTANT! Before you redirect a folder, confirm that no jobs are running on the same server (**Status > Running Jobs**).

To redirect the Shared Public folder:

1. Go to **Setup > Folder Locations**.
2. In **Shared Public Directory**, change the path so it points to the alternate directory.

Tips:

- To view the default path, click **Show Default**.
- To verify that the alternate directory is valid, click **Validate Locations**.

3. To save the new directory setting, click **Save**.
4. To enable the new directory on the server:
 - a. Immediately stop Pipeline Pilot.
 - b. Keep existing folders and files by copying subfolder contents from the previous location to the new directory.

Default directory location:
<installation-path>/public
 - c. Restart the server.

Redirecting the Local Temp Folder

A Local Temp directory is useful for managing local temporary storage. It can locate temporary job files on a file system that is local to the running job.

A Local Temp directory is advantageous when the server directory is located on a shared resource such as an NFS mount. It can improve protocol performance and reduce network loads on cluster and grid systems. When running some applications in this manner, administrators might need to redirect this folder to a local path that is valid for each node of the cluster. Each job running on the cluster will then create a job-specific subfolder for temporary files, accessible in the protocol by the global variable "LocalJobTempDirectory". Note that job result files should not be written to this location, which is cleaned up at the end of each job.

By default, this temporary file folder is the same as the main jobs folder. Temporary files are all written to the standard job temporary file subfolder, (by default, the global variables "LocalJobTempDirectory" and "TempDir" indicate the same folder location.)

IMPORTANT! Before you redirect a folder, confirm that no jobs are running on the same server (**Status > Running Jobs**).

To redirect the Local Temp folder:

1. Go to **Setup > Folder Locations**.
2. In **Local Temp Directory**, change the path so it points to the alternate directory.

Tips:

- To view the default path, click **Show Default**.
- To verify that the alternate directory is valid, click **Validate Locations**.

3. To save the new directory setting, click **Save**.

Redirecting the Upload Folder

The Upload folder is a directory that stores files uploaded to the server. This is an optional directory that administrators can configure if necessary. If an Upload directory is not defined (and the setting is left blank in the Admin Portal), User folders are used for this purpose.

IMPORTANT! Before you redirect a folder, confirm that no jobs are running on the same server (**Status > Running Jobs**).

To redirect the Upload folder:

1. Go to **Setup > Folder Locations**.
2. In **Upload Directory**, change the path so it points to the alternate directory.

Tips:

- To view the default path, click **Show Default**.
- To verify that the alternate directory is valid, click **Validate Locations**.

3. To save the new directory setting, click **Save**.

Redirecting the XMLDB Folder

The XMLDB directory stores protocols and components specific to the server. Some of these are public (Components and Protocols subfolders), and some are owned by specific users (Users subfolders). Besides a backup strategy for this data, you can relocate the directory to a more appropriate location for these valuable resources.

Preparing to Redirect an XMLDB Folder

The XMLDB directory is essential to client-server operations. The subfolders in this directory contain the indices for package components needed to run all protocols.

It is not okay to share an XMLDB folder between different server installations. Each installation requires a unique XMLDB directory to prevent version and file corruption problems.

Make sure you thoroughly read the following instructions before you redirect your XMLDB folder, as these operations have the potential to corrupt your protocol database if not properly followed.

To redirect the XMLDB folder:

IMPORTANT! Before you redirect a folder, confirm that no jobs are running on the same server (**Status > Running Jobs**).

1. Go to **Setup > Folder Locations**.
2. In **XMLDB Directory**, change the path to the alternate directory.

Tips:

- To view the default path, click **Show Default**.
- To verify that the alternate directory is valid, click **Validate Locations**.

3. To save the new directory setting, click **Save**.
4. To enable the new directory on the server:
 - a. Immediately shut down Pipeline Pilot.
 - b. Copy the "Components" and "Protocols" subfolders from the previous location to the new directory.

Default XMLDB directory location:

<installation-path>/xmldb>

Leave the "Objects" subfolder in the original directory (do not relocate it). This subfolder must remain in the default XMLDB directory. It contains basic configuration data for the server startup (e.g., all Folder Locations settings). For this reason, it needs to stay in the original directory.

- c. To retain existing XMLDB User tabs (applicable if this is not a fresh server installation), copy the "Users" subfolder from the previous location to the new location.
- d. Copy the folders with the same user name under which the Pipeline Pilot Apache server is running. (On Windows, see the Log In property page in the Services Control Panel.) The Apache user must overwrite XMLDB files, so ensure there are adequate file permissions.
- e. Restart the server.

Global Properties

Several global protocol properties are used to identify protocol names, user names, and the location where protocols run. You can also configure the server to use other global protocol settings on the global property list for each protocol it runs. For example, you can define contact information for internal tech support at your organization.

To add a new global protocol property:

1. Go to **Setup > Global Properties**.
2. In the **Custom Global protocol Properties** table, in the bottom, empty row enter the name of the global **Property** (for example, "CompanyName").
3. For the corresponding **Value**, enter the default value for the property (for example, "XYZ Research Corp.")
4. Click **Add**. The property is added to the global protocol property list.

To edit a global protocol property:

1. Select the **Property** you want to edit.
2. Change the name of the **Property** and the **Value** as required.
3. Click **Update**.

To remove a global protocol property:

- From the list of global protocol properties, select the one you want to remove, and click **Remove**.

Job Settings

You can customize how protocol jobs run on your servers. Default settings are applied during installation and you can modify them to better support your hardware and software configurations. The job performance settings that you can change are described below.

To change a job performance setting:

1. Go to **Setup > Job Settings**.
2. For each setting you need to modify, change the value.
3. Click **Save**.

| Setting | Description |
|---------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Running Job Limit | To prevent server overload, this option limits the number of jobs running on the server at the same time. The default maximum number of jobs that can run simultaneously is 10. Any excess jobs are placed in a queue and processed in order of submission, as running slots become available. To disable this job queuing feature, set the value to "0". |
| Block Job Timeout | Length of time (in seconds) before blocking jobs time out and return an error. The default is 10 seconds. |
| Job Priority Switching | When enabled, job processes are downgraded to a lower priority after 10 seconds. This feature is enabled by default. Normally, this setting does not significantly impact performance. |
| Intermediate Web Port Jobs Release Delay (seconds) | The time to delay the removal of intermediate Web Port jobs after they are run. All jobs that generate forms displayed in the left Web Port pane as well as jobs launched from the right Web Port pane are considered "intermediate". The value suffix can be "d" for days, "h" for hours or "m" for minutes. Set this to -1 or 0 to delay the removal of intermediate Web Port jobs forever. Note: The Pipeline Pilot administrator will need to clean up permanently delayed jobs. |
| Maximum Archived Jobs per Pipeline Pilot User | The server will actively delete jobs that exceed this maximum. Jobs that have an explicit or implicit expiration are excluded from the count. This includes blocking jobs and jobs that have been set in the client to expire. |
| Maximum Number of Simultaneous Parallel Processing Subprotocol Jobs | Maximum number of simultaneous jobs that can be spawned to support each individual parallel processing subprotocol. When the server hosts a parallel subprotocol operation, this setting is used by the running protocol to handle x-number of subprotocol jobs for a specific subprotocol. The default number of jobs is four (4). Increase the value based on the size of your machine and the number of jobs you need to run. |

| Setting | Description |
|-------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Maximum Number of Persistent Job Daemons per Job Pool | Maximum number of persistent daemons associated with a particular job pool. A single job or a set of related jobs can use daemons from the same job pool. After a pooled job finishes running, a daemon remains resident in memory for a configurable period of time waiting for new job requests. Running a job in a daemon is much faster because it skips initialization of many internal data structures. When the server is using a high amount of memory, no new daemons are created. The default maximum number of persistent daemons per job pool is 16. No daemons are created when the setting is set to 0. |
| Persistent Daemon Timeout | Maximum time (in seconds) that an idle daemon will remain resident in memory before it is shut down. The default timeout is 300 seconds (or 5 minutes). |
| Job Readiness Refresh Rate | To keep the server warm during periods of inactivity, a single protocol runs periodically to keep system files in memory and improve performance when starting new jobs. This manifests as an additional scisvr process. The configuration time, in seconds, sets the amount of time in between each protocol execution. The default is 300 seconds (or 5 minutes). To disable this feature, set the value to 0. |
| Pre-started Daemons for Non-Pooled Jobs | (Windows only) When set to a number > 0, a pre-initialized daemon is used to run jobs that are not assigned to any job pool. |
| Use Braces in Job Directory Names (Compatibility) | Generates braces for job directory names (to support backwards compatibility with previous server versions). If you have grids or clusters deployed on a Linux environment, the use of braces in paths may cause problems with third-party software. If your configuration uses grid engines, we recommend setting this option to "No". |
| Maximum Memory Usage of an Individual Job Process | Maximum amount of memory allocated on the server for processing jobs. All scisvr executables that exceed this limit are killed. Valid values include a percentage of total physical server RAM (for example, 75%), and raw number of gigabytes (for example, 16). |
| Maximum Job Age Based on Job Folder Size | Clean up users' jobs based on job age and minimum folder size in KB, MB, and GB. You can add multiple rows to define the maximum age for different folder sizes. Jobs that are set to expire at a future date are ignored. Cleanup is scheduled every four hours. |

Tips:

- You can specify where on the file system the job directories are created (go to **Settings > Folder Locations > Jobs Directory**).
- On clusters and grids, you can improve performance by allowing a job on a local disk to store temporary files (go to **Settings > Folder Locations > Local Temp Directory**).

Note: When using this option all temporary files will be removed when the job completes.

- Some settings may not apply due to your hardware and software configurations.
- For more details on customizing job settings to optimize your enterprise deployment, see "Server Performance Tuning" in the *Pipeline Pilot System Requirements Guide*.
- By default, the server keeps up to 100 jobs for each client user (Pipeline Pilot Client). Older jobs are deleted when the client reaches its maximum allotment. To increase or decrease the maximum number of jobs maintained on the server, go to [Setup > Server Configuration](#), and change the setting for **Maximum Archived Jobs per Pipeline Pilot User**.
- Web Port does not remove client jobs. The Web Port help recommends that users remove older files no longer needed from the Jobs tab.

Proxy Settings

If your network includes a proxy server, it may prevent Pipeline Pilot from accessing external Internet services. This can create problems for clients who need to run Web Port or Client SDK examples, since they need to communicate with web services. It can also pose problems for custom protocols that need to access the Internet.

You can configure Pipeline Pilot to recognize your network's proxy server by identifying its name or IP address in the Admin Portal.

Note: If a Pipeline Pilot Client runs from the same server where proxy settings are configured, that client uses these same settings (i.e., if it is installed in such a way that it is co-located with the server).

To set up a proxy server:

1. Go to **Setup > Proxy**.
2. In **Proxy Server URL**, enter the name of the proxy server or its IP address.
3. Specify how you want to authenticate:
 - If the proxy server requires authentication, enter the appropriate **Username** and **Password**.
 - To employ the username and password of the user running the protocol, check **Pass through the user's credentials from the executing protocol**. (The user's credentials must match those required by the proxy server for this feature to work.)
4. Click **Save**.

Setting up a Proxy Server Exclusion List

Some servers, such as those on an intranet, do not require a proxy server. Pipeline Pilot can bypass the proxy and access those servers directly by adding specific URLs to an Exclusion List.

To add a URL to the exclusion list:

1. Check **Exclude the local server addresses**.
2. In **Edit Excluded URL**, enter the first URL on your exclusion list, and then click **Add**.

- Repeat for each URL that you want to exclude from the proxy server.

Tips:

- To add a group of servers to the exclusion list, identify a subnet by entering the common part of their IP addresses or domain names, such as ".myco.net" or "10."
- Do not use wild cards since they are not recognized by this notation.
- You can also use the CIDR specification such as "192.168.0.0/16". (For details, see https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing.)

To delete an excluded site from the list:

- Select the URL from the exclusion list, and then click **Remove**.

Server Deployments

Server Deployments Overview

By default, Pipeline Pilot is configured for a standalone deployment. There are several options for deploying Pipeline Pilot in other ways, including:

- [Port forwarding firewalls and reverse proxies](#)
- [Load balancing](#)
- [Clustering](#)
- [Distributed grid computing](#)

For information about how to install Pipeline Pilot to support these other deployments, see the following guides:

- Pipeline Pilot Server Installation Guide
- Pipeline Pilot System Requirements Guide
- Pipeline Pilot Deployment Guide

Configuring Pipeline Pilot to Support Advanced Deployments

After installing Pipeline Pilot, you can configure features in the Admin Portal to support a particular deployment method. Review the following checklists before you make any changes in the Admin Portal. It may be necessary to perform a few preliminary tasks depending on your operating system and planned deployment.

Linux Checklist

| Do this: | Details: | Done? |
|--------------------------------|---------------------------------------------------------------------------------|--------------------------|
| 1. Shut down the Linux server. | <code>cd /opt/Accelrys/installdir/linux_bin</code> <code>./stopserver</code> | <input type="checkbox"/> |

| Do this: | Details: | Done? |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| 2. Run <code>scirootinstall</code> on Linux. | <p><code>scirootinstall</code> is located in the <code>apps/scitegic/installer/scripts</code> directory.</p> <p>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, adds <code>init</code> scripts, sets up <code>pam</code>, and configures the server to run with impersonation.</p> <div> Note: If you are setting up a cluster, <code>scirootinstall</code> should be run on all nodes in the cluster. </div> | <input type="checkbox"/> |
| 3. Set the group bit on directories on Linux. 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. | <p>Since <code>scirootinstall</code> does this for the default installation, this step should only be necessary if you use an alternate location.</p> <pre>mkdir AltUser chown ppuser.ppgroup AltUser chmod 2775 AltUser</pre> | <input type="checkbox"/> |
| 4. Run <code>mkpublic</code> on Linux. This is required when using a shared public directory in load balanced configurations. | <p>The <code>mkpublic</code> script is in the <code>apps/scitegic/installer/scripts</code> directory and can be run either as root or the server user. This command creates a compatible public directory with the correct permissions. Usage: <code>mkpublic [-u user] [-g group] path</code></p> | <input type="checkbox"/> |
| 5. For a grid, run <code>scigridsetup</code> on each node that will run protocols. | <p><code>scigridsetup</code> is located in the <code>apps/scitegic/installer/scripts</code> directory.</p> <p>This script adds an <code>init</code> script for <code>Xvfb</code> and runs <code>sciseallow</code>.</p> <div> IMPORTANT! If you do not run <code>Xvfb</code>, components that use R software to generate plots and graphics may not work. For further details, see the <i>R Software Install and Configuration Guide</i>. </div> | <input type="checkbox"/> |

Server Configuration Checklist

| Do this: | Details: | Done? |
|--------------------------------------------------------------------------------|-----------------------------------------------------------------|--------------------------|
| 1. Restart the Linux server if you previously stopped it to perform any tasks. | <pre>cd /opt/Accelrys/installdir/linux_ bin ./startserver</pre> | <input type="checkbox"/> |

| Do this: | Details: | Done? |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| 2. Log into the Admin Portal and configure the options that you want to use. | Features are available for deployment configuration, server security, job management, XMLDB maintenance, etc. | <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 (Security > Authentication). | <input type="checkbox"/> |
| 4. 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 the recommended clustering method. | <pre>ssh node1 cd /opt/Accelrys/installdir/linux_ bin ./startserver</pre> | <input type="checkbox"/> |
| 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 (Maintenance > Export Import Configuration). | <input type="checkbox"/> |

Reverse Proxy Deployments

Reverse Proxy Features

Port forwarding is the process of intercepting traffic bound for a certain IP/port combination and redirecting it to a different IP and/or port. This redirection may be accomplished by an application running on the destination host, or it may be performed by intermediate hardware, like a router, reverse proxy server, or firewall.

A reverse proxy server is a backwards proxy server. It acts as an intermediary for clients that want to access a web site by forwarding requests. A reverse proxy can lower the server's workload by redirecting requests to other similar servers via load balancing. Reverse proxies can also terminate SSL connections, which offloads SSL processing from the primary servers.

Note: Reverse proxies and port forwarding firewalls are supported on both Windows and Linux. Due to the differences in paths, only one type of operating system can be deployed across the enterprise.

Guidelines

The primary requirement for port forwarding is that the forwarder preserves the original host header from the HTTP request. This also applies to load balancers. For most hardware implementations, this is not an issue as hardware should be protocol-independent. When using Apache as a reverse proxy, it's necessary to edit the configuration to ensure that it preserves host headers.

Configuring a Reverse Proxy

When configuring for operation behind a reverse proxy, use a single port on the reverse proxy (SSL only) and terminate the SSL connection at the reverse proxy. This applies to any server running behind a port translating firewall, reverse proxy, or load balancer. (For details, see [Configuring a Single Port Operation](#).)

To configure support for a reverse proxy:

1. Go to **Setup > Reverse Proxy and Load Balancing**.
2. In **Full Name**, enter the fully qualified domain name for your reverse proxy.
3. In **Aliases**, enter any alias names for the reverse proxy, such as unqualified host or IP address.
4. In **Reverse Proxy Ports**, specify the ports that will be forwarded to your Pipeline Pilot Server.
5. To use SSL only, leave the **HTTP Port** field blank.
6. If you are configuring for load balancing, check **Load Balanced**.
7. Click **Save**.

Reverse Proxy and Load Balancing

The settings below tell the Accelrys Enterprise Platform (AEP) Server about your reverse proxy or load balancer. If you are configuring your server to work behind a reverse proxy, port translating firewall, or a load balancer, you will need to adjust these settings for your configuration. To modify a setting, change its value and then click **Save**.

Reverse Proxy Name and Aliases

Set the host name for your reverse proxy. The 'Full Name' should be the fully qualified host name for the reverse proxy, for example: `aepserver.mydomain.com`. The alias list should contain a semicolon separated list of names or IP addresses that the reverse proxy can be referred to, for example: `aepserver;192.178.21.7`.

| | |
|-----------|-----------------------------------------------------|
| Full Name | <input type="text" value="aepserver.mydomain.com"/> |
| Aliases | <input type="text" value="aepserver;192.178.21.7"/> |

Reverse Proxy Ports

Specify the ports that users will use to connect to your reverse proxy. If you wish to provide SSL only access you can leave the HTTP port field blank or vice versa.

| | |
|-----------|-----------------------------------|
| HTTP Port | <input type="text"/> |
| SSL Port | <input type="text" value="9003"/> |

Load Balancing

Check the box below if your servers will be behind a load balancer. Please refer to the administration guide for further setup instructions to configure your server for use behind a load balancer. **Enabling this feature will apply some restrictions on functionality.**

☒ Load Balanced

Load Balancing Deployments

Load Balancing Features

Load balancing distributes incoming HTTP requests across web servers in a server farm, to avoid overloading any one server. Because load balancing distributes the requests based on the actual load at each server, it is excellent for ensuring availability and defending against denial of service attacks.

Notes:

- If you have an enterprise deployment and have a BIOVIA Foundation Hub server, follow the instructions in the *BIOVIA Foundation Hub Deployment Guide* for configuring load balancing.
- Load balancing is supported on both Windows and Linux. Due to the differences in paths, only one type of operating system can be deployed across the enterprise.

Guidelines

Pipeline Pilot should work with any hardware or software load balancer. The Pipeline Pilot software needs to be installed on each node that will be used as part of the load balanced cluster. The software must be installed in the same path for every node. Each node is independent and only the user, jobs, and public directories are shared. These directories do not need to be local to any node.

Pipeline Pilot fully supports request-level load balancing. The load balancer must preserve the original host headers for all requests. Since each request can go to a different server, there are no requirements for affinity; however, applications built on the platform may require it.

Configuring for Load Balancing

With a load balanced deployment, be aware of the following limitations:

- Under load balancing, the XMLDB is read-only. It is not possible to save protocols or create new folders in the XMLDB. Additionally, some model-building protocols are not operational. (Protocol updates must be handled through packages using pkgutil. For further details, see the *Application Packaging Guide* (go to **Pipeline Pilot Help Center > Developers tab > Development Guides**).
- To deploy protocols on a load balanced server, it is necessary to use the packaging system and add the protocols to the servers.
- All server installations should have the same set of packages installed. If different packages are installed on different servers, your users will be directed to specific servers and will not have package access.

| Do this: | Details: | Done? |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| 1. Perform the steps for configuring a reverse proxy . | This applies to servers running behind a load balancer (hardware or software). | <input type="checkbox"/> |
| 2. After configuring a load balanced server to your specifications, export the server settings and apply them to other load balanced servers in your cluster. | <p>To reuse your configuration on other servers, go to Maintenance > Export Import Configuration. For further details, see Exporting and Importing Server Configurations.</p> <div> <p>Note: When load balancing is configured, it is not possible to configure server settings from the load balancer. Instead, access each node individually.</p> </div> | <input type="checkbox"/> |
| 3. Set the shared locations for the User Directory, Jobs Directory, and Shared Public Directory. | Go to Setup > Folder Locations . For further details, see Managing Folder Locations . | <input type="checkbox"/> |
| 4. (Optional) To improve performance, you can also set the Local Temp Directory (Setup > Folder Locations). | The temporary data on the local temp directory will not be accessible by any server other than the one running the job. For further details, see Redirecting the Local Temp folder . | <input type="checkbox"/> |

Distributed Grid Computing Deployments

Grid Features

Grid computing is a type of distributed computing where CPU resources are shared across a computer network, allowing all connected machines to operate as a compute cluster.

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 farmed out 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 group of clients to run more protocols with maximum efficiency for each individual job.
- With parallel computing, you can efficiently distribute individual jobs to cluster nodes, making it possible to tackle extremely large data sets that were previously off limits, enhancing the client's parallel processing capabilities.
- You can run any subprotocol in parallel on a specified set of remote servers and specify data record batch size, 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. You can also run normal protocols under the grid engine.

Note: Distributed grid computing is only supported on Linux.

Support for Grid Engines

A grid installation is virtually identical to a cluster installation. The only real difference is that instead of running Pipeline Pilot on the cluster nodes, the grid engine software manages the cluster nodes.

Installation

For information about installing grids with Pipeline Pilot, see the *Pipeline Pilot Installation and Configuration Guide*.

Guidelines

The Pipeline Pilot software only needs to be installed once and is shared across all nodes. The path to the installation must be the same for all nodes.

Note: For details about required grid engine software supported on the platform, see the *Pipeline Pilot System Requirements*.

Configuring for Grid Operation

Grid computing distributes CPU resources across a computer network, allowing all connected machines to operate as a compute cluster. A Grid page is available in the Admin Portal to configure and manage grids (Setup > Grid). From here, you can specify the type of grid engine software you are running, so Pipeline Pilot knows which script it should use to submit, monitor, and cancel grid jobs.

To configure your server for grid operation:

1. Go to **Setup > Grid**.
2. Set Grid Engine Configuration options:
 - **Grid Engine Type:** Select the grid engine (SGE, PBS, LSF, and CUSTOM).
 - **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 actually number of grid job submissions will probably be more that 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**.

Tips:

- You can configure the grid operation behind a [reverse proxy](#) or [load balancer](#).
- 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](#).
- Local temporary storage can improve protocol performance and reduce network loads on clusters and grids.

Grid Engine Queue Names

You can expose queue names in the Pipeline Pilot Client interface. Client users can select a queue name for each protocol that runs on the grid. Type a name in the **Queue Name** text box and click **Add Queue**. Click the red X to remove an entry.

Testing a Grid Engine with Pipeline Pilot

1. In Pipeline Pilot Client, ensure you are connected to the Linux server that is properly configured for grid engine support.
2. Open the example protocol *Grid Example 1* or create a simple protocol that you can run on this server.
3. At the protocol level, set the *Run On Grid* parameter to "True". (This is already set in the example protocol via the Implementation tab.)
4. If you added grid engine queue names in the Admin Portal (Setup > Grid), they are available as *Queue Name* parameter options in Pipeline Pilot. Open the *Run on Grid* parameter group and select the queue that should run your job. (If you do not specify a queue name, the grid engine default is used.)

NOTE: This protocol requires the server to be configured to use a grid engine. All protocols will still run without a grid engine configured, but you won't be able to see the grid engine support without it.

This protocol demonstrates two methods for using grid engine support. You can enable the entire protocol or just subprotocols for grid operation. In some cases you may want to only run part of the protocol on the grid, alternatively your site may require you to run all your jobs on the grid. Pipeline Pilot allows you to do either or both.

Elapsed Time:

Grid Example 1*

| Implementation | |
|-------------------------------------------------|-------------|
| Tempfiles | |
| Protocol Form | |
| <input checked="" type="checkbox"/> Run On Grid | True |
| Queue Name | |
| | ibmblade_q1 |
| | ibmblade_q2 |

Parameters Implementation Web Service

Clustering Deployments

Clustering Features

A cluster is a set of loosely connected computers that work together to function like a single system. Each computer used like a server is known as a "node" and runs its own operating system. Clusters are usually deployed to improve performance and availability over that of a single computer, while typically being much more cost-effective than single computers of comparable speed or availability.

Clusters are configurations where cluster-nodes share computational workloads to provide better overall performance. Pipeline Pilot clustering allows you to scale your server to take advantage of multiple servers for distributing client jobs. The primary server, or head node, acts as a dispatcher to the other servers in the cluster. Each node in the cluster includes a Pipeline Pilot Server and shares the installation with the head node.

Note: Clustering is only supported on Linux.

Installation

For information about installing Pipeline Pilot for use with clustering, see the *Pipeline Pilot Installation and Configuration Guide*.

Clustering Modes of Operation

Pipeline Pilot clustering has two basic modes of operation:

- **Private mode:** The client connects to the head node and all jobs are distributed to the child nodes by the head node on a per-job basis. In the private mode, requests are proxied to the runner nodes by the head node, the client does not connect to the runner nodes.
- **Public mode:** The client initially logs into the head node and is assigned a new child node for running each job. In public mode, the client contacts the runner nodes directly.

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

Support for Clustering

When a client requests to run a protocol, the running protocol and its resulting data are known as a "job". Clustering is a technique that provides scalability for jobs by efficiently using available network resources. Jobs requested by a client that are connected to a given server are delegated to other computational nodes. The cluster is defined by the set of nodes available for delegation.

Clustering capabilities are available for Pipeline Pilot Servers that run on the Linux platform. In this configuration, a single Pipeline Pilot installation on a shared file system is accessed from multiple server machines. They share the same Pipeline Pilot installation file system and the same protocol database (XMLDB). The only unique aspect to each server node is the specific set of jobs assigned for execution.

Note: The configuration for grid engine integration is virtually identical to the configuration for clustering. This guide provides details for each.

Guidelines

A Pipeline Pilot cluster consists of a set of compute nodes that share a mounted file system upon which you install the Pipeline Pilot software.

Pipeline Pilot clustering requires the following:

- One node should be designated as the primary node. All other nodes are treated as 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.
- All nodes must have access to the shared file system where you install Pipeline Pilot.
- The path to the Pipeline Pilot installation must be identical for all nodes. (For example, the path `"/mnt/shared/scitegic"` is valid for all nodes, primary and secondary.)
- All client machines running client software should have HTTP access to the primary node. If client machines do not have access to the secondary nodes, you can set a "Private Cluster" option in the Admin Portal (Setup > Clustering, explained below).
- To request web services, all nodes should have HTTP access to the primary node, and the primary node should have HTTP access to all secondary nodes.
- 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 lets you specify a location that must exist on all nodes with RWX (*Read, Write, 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. (See the *Pipeline Pilot System Requirements Guide*.)

Node Maintenance

It is necessary to maintain a list of secondary nodes for the primary node in the Admin Portal. This list is used to delegate jobs according to a clustering algorithm. There are two methods available in the Admin Portal for maintaining nodes. One is a simple round-robin technique, and the other attempts to maintain a level number of jobs running in each node. Ensure that an instance of the Pipeline Pilot software is running on each node in the cluster. If one node is out of action at any point, the primary node selects a different node, so the node list does not need to be maintained on a minute-by-minute basis.

Configuring for Clustering

Server clustering provides load balancing scalability for running jobs. It optimizes the use of available network resources on Linux. A Clustering page is available in the Admin Portal to manage server cluster settings (Setup > Clustering).

To configure your server for clustering:

1. Go to **Setup > Clustering**.
2. From the **Load Balancing** list, select an option:
 - **Round Robin:** Jobs are assigned to each server in sequence.
 - **Job Leveling:** Servers are prioritized with the fewest jobs.
3. 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.
4. 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.

5. 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** and then click **Add**.

Admin Pages << + -

- Home
- Maintenance
- Reports
- Security
- Setup
 - Data Sources
 - File Browsing
 - Folder Locations
 - Global Properties
 - Job Settings
 - Proxy
 - Reverse Proxy and Load Balancing
 - Server Configuration
 - Server Registry
 - Validation Rules
 - Clustering**
 - Grid
- Status

Clustering

Clustered Server Management

(Linux only) You can create a cluster of servers, each with its own instance of the server running from a single NFS-mounted Accelrys Enterprise Platform (AEP) Server installation. Jobs sent to the server will be distributed across the different machines in the cluster. For more information, see the help page or contact Accelrys Technical support. Note that all clustering is disabled if the *Load Balancing* setting below is set to Off.

Load Balancing: Round Robin (Job assigned to each server in sequence) ▼

Internal Server Name

If you are configuring for clustering or a grid engine, you must specify the internal server name to be used to contact the AEP head node from within the cluster. This allows you to take advantage of local high speed back end networks for clusters and grids. This host name must be visible from inside the network segment for the cluster or grid. You may use an IP address in this field if desired. **This setting is required for clusters and grids.**

Internal Server Name: Save

☒ **Private Cluster** (Endpoints are accessible from the primary server, but not from client machines).

Cluster Runner Endpoints

To add a server to a cluster, enter the server name in the text box below and click *Add*. In a cluster, all the endpoint servers run using the same port, since they are all based on a single NFS-mounted installation.

| Active? | Endpoint | Parallel Subprotocols Only | Remove |
|----------------------|--------------|---------------------------------------------------------------|--------|
| | campus_point | <input type="radio"/> Yes <input checked="" type="radio"/> No | |
| <input type="text"/> | | | Add |

- (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.



Server Configuration

Configuring Pipeline Pilot Servers

A variety of settings define how your server runs and interacts with clients, databases, web services, and other applications and features. All default settings are configured at installation time. You can modify these settings whenever necessary.

To change a server setting:

1. Go to **Setup > Server Configuration**.
2. Find the setting you need to change (defined below) and choose a setting.
3. Click **Save** to update the server.

| Setting | Use this feature to: |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XMLDB Endpoint | Share an XMLDB with multiple servers by specifying the name of a host XMLDB. By default, the XMLDB for the logged on server is used. Default: %httproot%/scitegic/xmlldb See Sharing an XMLDB with Multiple Servers |
| Automatic Client Download | Enable the server to prompt Pipeline Pilot Client users to download and install a new client version if the currently installed client version is incompatible with the server. Default: Yes See Client Support |

| Setting | Use this feature to: |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| File Browsing | <p>Control what server folder locations are displayed in client file browsers.</p> <ul style="list-style-type: none"> ■ Unrestricted: Users can browse for files anywhere on the server. ■ Restricted: Users can only browse within the "scitegic/public" directory. ■ User folder only: Users can only browse within their own user folder. <p>Note: Impersonation settings can also control access to folders by individual users.</p> <p>Default: Restricted See Adding File Browser Shortcuts</p> <p>Recommendation: Leave Restricted. Users can only browse within the scitegic/public directory tree. When set to User folder only, users can only browse within their own user folder. Note that Impersonation as set from the security configuration pages can also control access to folders by individual users.</p> |
| File Browsing > Read Access Flags | <p>Flags to modify read access rights to specific locations, used in combination with the File Browsing mode. Flags which are underlined are those enabled by default for the current browsing mode.</p> <ul style="list-style-type: none"> ■ Package Shortcuts: Package locations declared as data folders by the package author. ■ UNC Paths: (Windows) Resource location, beginning with double backslashes. For more precise control, a specific UNC path shortcut can also be defined by an Administrator. ■ Administrative Shares: (Windows) Hidden share of a logical hard drive where the drive letter has a '\$' suffix. ■ Jobs Folder: Top level jobs folder. ■ Others' Content: User data folders and job folders of other users. This might be appropriate in a collaborative environment where data security is not a concern. |

| Setting | Use this feature to: |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| File Editing | <p>Control the folder locations where client file browsers can upload or modify files.</p> <ul style="list-style-type: none"> ■ Unrestricted: Users can delete, move, and rename files and directories outside of their own "scitegic/public/users" file directory. ■ Restricted: Users can only modify files and directories within their own user directory. <p>Note: This setting is ignored when impersonation is enabled, since the user's own file system permissions are used instead.</p> <p>Default: Restricted See Adding File Browser Shortcuts</p> |
| Unrestricted File Download | <p>Control the directories that are accessible without any authentication, as a semicolon-separated list.</p> <p>Default: [no setting, blank]</p> <p>Recommendation: <i>Leave empty.</i> This ensures all file download requests are authenticated.</p> |
| Job Directory Access | <p>Restrict job directory web access to the owner. When restricted, only the user that ran the job can access files in job directory via the web server interface.</p> <p>Default: Unrestricted</p> <p>Recommendation: <i>Change to Restricted.</i> This ensures only the user that ran a job can access the result files.</p> |
| Maximum Number of CPUs to Use | <p>Set the maximum number of logical CPUs that Pipeline Pilot can use. Logical CPUs are those detected by the operating system, independent of whether that CPU is implemented as a physical processor, a multi-core processor or with hyperthreading.</p> <p>Default: [no setting, blank]</p> <p>Note: To apply this setting, either restart the server or wait for a few jobs run, so that cached processes using the old setting are cleared out.</p> <p>See Recommendations for CPU Usage.</p> |

| Setting | Use this feature to: |
|-------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Expiration (days, hours or minutes) of Single Sign-on Credentials | <p>Set the inactivity timeout before credentials expire in the context of web browser-based-logon. Users must log on again, if their credentials expire due to inactivity. If a user logs on again before the expiration, the credentials are renewed and the inactivity clock is reset.</p> <p>Default: 30d</p> <p>Recommendation: <i>Change to 30-60 minutes</i>, which is recommended for web application usage.</p> <p>See Support for Security Issues.</p> |
| Retain Session Cookie beyond Web Browser Session | <p>Control how the logged on browser handles cookies.</p> <ul style="list-style-type: none"> ■ Yes: Sets an expiration on the session cookie, so that the cookie will be saved to disk until the time of expiration. ■ No: The cookie will not exist after the web browser session is closed. <p>Default: Yes</p> <p>See Support for Security Issues</p> <p>Recommendation: <i>Change to No</i>. This forces users to remove session cookies when the browser closes. This means users must provide their credentials again when opening a new browser session.</p> |
| Enable HttpOnly flag on Session Cookies | <p>Determine whether session cookies will use the HttpOnly flag. This will mitigate the effect of cross-site scripting attacks but prevent JavaScript-based applications from accessing the cookie value.</p> <p>Default: No</p> <p>See Support for Security Issues</p> |
| Validate Redirect URLs | <p>Specify whether redirect URLs will be validated before authentication, this ensures that users will not be forwarded to untrusted sites.</p> <p>Default: No</p> |
| Session Salt | <p>Assign an additional password to encrypt contents of the session.</p> <p>Default: [no setting, blank]</p> <p>See Support for Security Issues</p> |
| Compress XMLDB | <p>Control how protocol and component files are saved in the XMLDB.</p> <ul style="list-style-type: none"> ■ Yes: Reduces the space occupied on your server for XMLDB file storage. (Compressed files are saved with the extension "xml.gz".) ■ No: Saves all files in the XMLDB at their default uncompressed size. (This setting requires more disk space on your server.) <p>Default: Yes</p> <p>See Compressing an XMLDB</p> |

| Setting | Use this feature to: |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Keep Alive | <p>Allow an HTTP connection to be re-used for several requests within a given timeout limit. This avoids the need to create a new connection between client and server for every request.</p> <p>The time-limit value is calculated automatically by Pipeline Pilot depending on the server's capabilities. It can vary from 10 s up to a maximum of 40 s.</p> <p>IMPORTANT! Only set Keep Alive to "No" if you are experiencing stability problems with clients (for example, if some .NET clients do not handle Keep Alive correctly).</p> <p>Default: Yes</p> |
| Compress HTTP Messages | <p>Enable mod_deflate for Apache. (It is necessary to restart the server if this setting is changed.)</p> <p>Default: Yes</p> <p>See http://httpd.apache.org/docs/current/mod/mod_deflate.html</p> |
| Enable Server Info | <p>Control whether potentially insecure services are enabled in the Apache process. If you change this setting, you must restart Apache, see Restarting the Server.</p> <p>Default: Yes</p> <p>Recommendation: Change to No. This ensures sensitive information cannot be accessed from outside the Admin Portal.</p> |
| Allow Cross-origin Requests | <p>Define one or more origin endpoints for servers that are allowed to make cross-origin HTTP requests to this Pipeline Pilot Server. List multiple allowed origins (host name and port, if not default) with spaces between endpoints, the wildcard * is supported.</p> <p>For example: appsrv.intra.net:1944 *.5ds.net</p> <p>This uses the Cross-Origin Resource Sharing (CORS) standard, by adding specific HTTP headers in a response to indicate when a named Origin server is allowed to access resources on the Pipeline Pilot Server.</p> <p>Default: [no setting, blank]</p> |
| Server Port Usage | <p>Configure Pipeline Pilot for single port operation (typically applies to standalone server installations).</p> <p>Default ports used for Pipeline Pilot and Apache web server are:</p> <ul style="list-style-type: none"> ■ HTTP Only: Primary server port. Always in use and supports Web Port, Pipeline Pilot Help Center, and all non-secure web services. ■ SSL Only: This port is managed by a security protocol (SSL). It is always in use and supports the Admin Portal and the locator web service that includes the user login procedure. <p>Recommendation: Change to SSL Only. This ensures all traffic is passed over a secure SSL/TLS connection and session cookie or security tokens cannot be intercepted and misused.</p> <p>See Configuring a Single Port Operation and Reconfiguring Ports</p> |

| Setting | Use this feature to: |
|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SSL Security Level | <p>Configure the SSLCipherSuite and SSLProtocol values of the Pipeline Pilot HTTPS port based on the recommendations from https://wiki.mozilla.org/Security/Server_Side_TLS.</p> <ul style="list-style-type: none"> ■ Intermediate: This is appropriate for services that need to support a wide range of clients and is compatible with Firefox 1, Chrome 1, IE 7, Opera 5, and Safari 1. ■ Intermediate (Java 6 support): This disables DHE based ciphers so that you can use Java 6 clients as these are unable to connect to servers with Diffie-Hellman parameters > 768 bits. ■ Modern: This only allows TLS 1.2 for services that do not need to support older clients. This configuration is compatible with Firefox 27, Chrome 30, IE 11 on Windows 7, Edge, Opera 17, Safari 9, Android 5.0, and Java 8. Notably, .NET v4.5 and older clients that make use of WebRequest default to TLS 1.0 and must explicitly enable TLS 1.2 in the source code or the Windows registry. You should ensure that you test your own clients for compatibility if you select "Modern". <p>Default: Intermediate See SSL Security Level</p> |
| Allow Remote Administration | <p>Control remote access to the Admin Portal.</p> <ul style="list-style-type: none"> ■ Yes: Enables remote access. ■ No: Disables remote access. <p>Default: Yes See Remove Administration Access</p> |
| Allow Upload to ScienceCloud Exchange | <p>Specify whether to allow Pipeline Pilot Clients connected to this Pipeline Pilot Server to upload to the ScienceCloud Exchange server at exchange.sciencecloud.com.</p> <p>Default: Yes</p> |

Recommendations for CPU Usage

Pipeline Pilot runs well with two (2) concurrent jobs per one (1) processor core. Optimal performance is realized with one (1) concurrent job per core. This calculation helps you estimate your processor core requirements based on total CPU time needs of your actual users. To leave system capacity for operating system and other critical processes, set the maximum number of CPUs that can be used to 80 percent or less than the total available.

The optimal number of CPUs required for efficient running of a Pipeline Pilot Server depends on the types of jobs that are run, if queuing is enabled, job run frequency, and the number of concurrent users. For example, a web client application such as a registration database requires approximately four (4) CPU time per job and must support 150 concurrent users. The Pipeline Pilot Server has at least three (3) CPU cores to prevent queuing and delays in jobs being experienced by users.

To limit CPU usage:

1. Go to **Setup > Server Configuration**.
2. Enter a value for the **Maximum Number of CPUs to Use** according to your requirements.
3. Click **Save**.

Note: A Server Sizing Worksheet is available at the top of the download (bundled with installation guides). The worksheet provides more information to help you estimate an optimal configuration for your server setup. For additional details, contact Dassault Systèmes Support.

Remote Administration Access

By default you can connect to the Admin Portal for a Pipeline Pilot Server from any machine which has access to the server (for example, it is on the same network). You can limit access to the Admin Portal so that it can only be accessed from the server machine.

To enable/disable remote administration:

1. Select **Setup > Server Configuration**.
2. To disable access to the Admin Portal from remote machines, set **Allow Remote Administration** to "No".
3. To enable remote administration, set this to "Yes".
4. Click **Save**.

SSL Security Level

The *SSL Security Level* setting on the Server Configuration page of the Admin Portal controls the `SSLProtocol` and `SSLCipherSuite` directives for the Pipeline Pilot SSL configuration in apache.

The *SSL Security Level* setting can take three options (see [Configuring Pipeline Pilot Servers](#) for more details):

- **Intermediate:** This is appropriate for services that need to support a wide range of clients and specifies:
 - **SSLProtocol:** `all -SSLv2 -SSLv3`
 - **SSLCipherSuite:**
ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:
ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:
ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:
DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:
ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:
ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:
ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES256-SHA384:
ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:
DHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA256:DHE-RSA-AES256-SHA:
ECDHE-ECDSA-DES-CBC3-SHA:ECDHE-RSA-DES-CBC3-SHA:
EDH-RSA-DES-CBC3-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:
AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:DES-CBC3-SHA:!DSS

This supports TLSv1, TLSv1.1, and TLSv1.2. Ciphers are listed in order of preference

- **Intermediate (Java 6 support):** This disables DHE based ciphers so that you can use Java 6 clients. This uses the same **SSLCipherSuite** as the *Intermediate* setting, but disables all DHE based ciphers in order to allow Java 6 based clients to connect.

- **Modern:** This only allows TLS 1.2 for services that do not need to support older clients and specifies:

- **SSLProtocol:** `all -SSLv3 -TLSv1 -TLSv1.1`
- **SSLCipherSuite:**
`ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:`
`ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:`
`ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:`
`ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:`
`ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256`

This only supports TLSv1.2. Ciphers are as described in order of preference.

Configuring a Single Port Operation

By default, the Pipeline Pilot Server has two open ports. The secure port is only used for authentication. You can now configure your server for single port operation. This primarily applies to standalone server installations.

To configure Pipeline Pilot to operate on a single port:

1. Go to **Setup > Server Configuration**.
2. Set **Server Port Usage** to "SSL Only" or "HTTP Only".
3. Click **Save**.

Note: The opposite port will still allow a connection, but all traffic will be directed to the specific port selected in Server Port Usage. It is possible to completely disable the listener for the port you are not using. If you need to do this, contact Technical Support for instructions.

Reconfiguring Ports

Server ports are configured when Pipeline Pilot is installed on your server. You can verify the port numbers used in your configuring by going to **Status > Server Information**.

The default ports for Pipeline Pilot are:

| Port Type | Default Port Number |
|----------------------|---------------------|
| SSL Port | 9943 |
| HTTP Port | 9944 |
| Tomcat Shutdown Port | 9945 |
| Tomcat HTTP Port | 9946 |
| Derby Port | 9947 |

Note: If you need to change these port numbers, you can reinstall Pipeline Pilot or manually edit the "`<pps_install>/install/tokens.config`" file. This file contains all of the port settings that are used by the server (plus other vital configuration information that should be left as-is). Also, you will need to re-index the catalog. See [Catalog Settings](#).

To manually edit "tokens.config":

1. Open the "<pps_install>/install/tokens.config" file in a text editor.
2. The port settings that you can modify include:
 - **port:** The primary non-secure Pipeline Pilot Server port (HTTP).
 - **ssport:** The primary secure Pipeline Pilot Server port (SSL).
 - **javaserverHttpPort:** The primary Java server port (Tomcat HTTP).
 - **javaserverShutdownPort:** The port used by the Java server to listen for shutdown messages (Tomcat Shutdown).
 - **javaserverDerbyPort:** The port used for the local Derby database.
3. After modifying port numbers, save the file.
4. Shut down the Pipeline Pilot Server. (For further details, see [Managing Pipeline Pilot Services](#).)
5. On Linux servers, first execute the command "source ppvars.sh" (if using the bash shell), or "source ppvars.csh" (if using the C shell). This sets environment variables required for successful execution of DbUtil (see below).
6. Run `DbUtil -t` to update the server configuration files.

IMPORTANT! Errors that occur when running `DBUtil -t` are not returned to the command line. Check `rterror.log` under <scitegic_root> to verify the process ran successfully.

7. Reinstall the Java Server package, from the <pps_install>/bin or <pps_install>/linux_bin folder, run the command:

```
pkgutil -r scitegic/javaserver
```
8. For Windows, open a command window and navigate to <pps_install>\bin and run the following commands:

```
AEPManger.exe -k uninstall  
AEPManger.exe -k install
```
9. Restart the Pipeline Pilot Server.

Do not modify any other settings in "tokens.config". If you have questions, contact Dassault Systèmes Customer Support.

Settings for Web Client Hosts

If your Pipeline Pilot Server is used to host application access through web clients, certain settings can be configured to enhance the performance experienced by client users.

Note: Some additional security settings may also be of interest when configuring Pipeline Pilot as a host for web client applications. For further details, see [Support for Security Issues](#).

The following settings should be carefully configured according to your requirements:

| Setting | Admin Portal Location | Default | Recommended Value | Details |
|---------------------------------------------------------------------|---------------------------|---------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Authentication Method | Security > Authentication | None | Domain or Local | No authentication ("None") will allow any user to access the Pipeline Pilot Server and any applications it hosted without entering a username or password. |
| Impersonation | Security > Authentication | None | None | Activating Impersonation will adversely affect scalability. |
| Running Job Limit | Setup > Job Settings | 10 | $(2 \times \text{cores}) + 4$ | If too many jobs are running on not enough cores the performance and scalability will suffer. |
| Blocking Job Timeout | Setup > Job Settings | 40 | 40 | This is the time to wait for a job that <i>must</i> be completed to run, if this limit is reached an error will be returned. |
| Job Priority Switching | Setup > Job Settings | Yes | Yes | This prevents a normal job from using excessive resources and thus delaying or slowing other jobs. After 10 s the priority will be reduced to allow other jobs to run. |
| Maximum Number of Simultaneous Parallel Processing Subprotocol Jobs | Setup > Job Settings | 4 | $(1 \times \text{cores}) - 1$ | The recommended setting ensures that no matter how many cores are being consumed by a parallel job, there will always be one unused core available for other jobs. |
| Maximum Number of Persistent Daemons per Job Pool | Setup > Job Settings | 16 | $(2 \times \text{cores})$ | Use a value of 2 if Impersonation is set to "Full" or "Restricted". |
| Persistent Daemon Timeout (seconds) | Setup > Job Settings | 300 | 1800 | Use a value of 300 if Impersonation is set to "Full" or "Restricted". |
| Job Readiness Refresh Rate (seconds) | Setup > Job Settings | 300 | 300 | This keeps a job process running at all times so that when new jobs are run the start-up cost is reduced. |

| Setting | Admin Portal Location | Default | Recommended Value | Details |
|------------------------------|-----------------------|---------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Connection Timeout (seconds) | Setup > Data Sources | [empty] | 300 | This keeps connections to the database open for a limited time so that jobs which involve interaction with the database have lower start-up costs. |

Tip: [Clustering](#) is strongly recommended for servers running web applications with more than 10 concurrent, heavy users.

Server Registry

Use the Server Registry page to register the Pipeline Pilot Servers for the following purposes:

- Identify the servers to monitor in the [Server Usage report](#)
- Specify which XMLDBs to use with [Catalog Search](#).

To add a server to the registry:

1. Go to **Setup > Server Registry**.
2. In the text entry box, type the location of the server in the format "servername:SSLport" (e.g., mercury:9943).
3. Click **Add**.
4. In the panel that opens, enter the username and password for a user who can log into this server. If a username is not specified, your Admin Portal login credentials are used (your name is then displayed in "User name").
5. Click **Update**.

Tips:

- To remove a server from the registry, click **Remove**.
- To modify the settings for a server, click **Edit**, and then make your changes.
- "Last Log Update" indicates the last time a usage log was copied from the server (the log is copied whenever you create a usage report).

Validation Rules

A validation protocol that checks an existing protocol for a variety of errors can be run either from the Pipeline Pilot Client or from the Admin Portal. (For details, see [Validation Reports](#).)

This is especially useful when deploying a protocol from a test to a production server, where the destination environment may not be the same and could cause problems running the protocol.

A validation protocol is installed on your server. Client users are able to manually run the validation protocol on their "active server" connection. Administrators can use the Admin Portal to use a customized validation protocol.

To configure the validation protocol:

1. Go to **Setup > Validation Rules**.
2. In **Pipeline Pilot Validation Rules**, enter the location of the rules protocol for validation run from the Pipeline Pilot Client, relative to the XMLDB. The default path to the system-supplied validation protocol is "Protocols/Utilities/Validation/Protocol Validation Rules (Default)".
3. In **Administration Validation Rules**, enter the location of the rules protocol for validation run from the Admin Portal, relative to the XMLDB. The default path to the system-supplied validation protocol is "Protocols/Utilities/Validation/Protocol Validation Rules (Default)".
4. Click **Save**. After the validation rules protocol is changed, it goes into effect when client users restart Pipeline Pilot or reconnect to their active server. The change is available immediately in the Admin Portal.

To mandate validation for an operation:

1. For the operation which should always be validated (Save, Export, or Mail) set **Mandate Validation** to "Yes".
2. For each operation where validation required, specify what you want to do if there are errors:
 - To continue the operation, for **Action on Validation Failure**, select "Continue".
 - To cancel the operation, for **Action on Validation Failure**, select "Cancel".

To disable mandatory validation for an operation:

- For the operation which should not always be validated (Save, Export, or Mail) set **Mandate Validation** to "No".

Publication Targets

When a protocol is published from a Pipeline Pilot Client a target endpoint can be selected. These endpoints are defined for the non-package protocol folders (and their subfolders).

To define a publication target:

1. Go to **Setup > Publication Targets**.
2. In the empty row in the table specify:
 - **Name** - this is a simple identifier for the target
 - **Display Text** - this is the text displayed in the Pipeline Pilot Client
 - **Folder Scope** - this is folder (and its subfolders) for which the target will be available
 - **Endpoint** - the URL to open when the target is selected in the Pipeline Pilot Client
 - **BitMap** - an optional icon to display in the Pipeline Pilot Client for this target
3. Click **Add** to create a new publication target.

Tip: Click **Clear Form** to remove all the values from the new row.

To edit a publication target:

1. Select the **Publication Target** you want to edit.
2. Change the target's properties as required.
3. Click **Update**.

To remove a publication target:

- From the list of publication targets, select the one you want to remove, and click **Remove**.

Privacy Policy

The **Setup > Privacy Policy** page allows you to define and manage a legal notice, such as Personal Data Protection policy, that you require your users to accept before signing in. For details about using this feature, see the *BIOVIA Personal Data Protection Administration Guide*, which is included with the documentation available with the Pipeline Pilot installation files.

Remote Administration Access

By default you can connect to the Admin Portal for a Pipeline Pilot Server from any machine which has access to the server (for example, it is on the same network). You can limit access to the Admin Portal so that it can only be accessed from the server machine.

To enable/disable remote administration:

1. Select **Setup > Server Configuration**.
2. To disable access to the Admin Portal from remote machines, set **Allow Remote Administration** to "No".
3. To enable remote administration, set this to "Yes".
4. Click **Save**.

Chapter 6:

Server Maintenance

Setting up MongoDB

MongoDB Overview

MongoDB is an open source document-oriented database system that is part of the NoSQL family of database systems. Unlike traditional relational databases that store data in tables MongoDB stores structured data as BSON (JSON-like) documents with dynamic schemas (document-oriented storage). MongoDB was designed for ease of development and scalability, with particular emphasis on web applications and infrastructure. Document-oriented database capabilities more efficiently handle data integration.

MongoDB Components

A set of prototype components is available that can take advantage of MongoDB's capabilities, (e.g., you can use these components to develop your own applications.) These prototype components are available in "Components\Data Access and Manipulation\Utilities\Prototypes\MongoDB". For further information, see the component reference help.

Requirements

MongoDB is supported on multiple 32- and 64-bit platforms (e.g., OS X, Linux, Windows, Solaris). Support for 32-bit platforms is limited to 2GB of data (<https://www.mongodb.com/try/download/enterprise>). As a convenience, Pipeline Pilot includes the MongoDB redistributables (version 2.6.6) for Windows 64-bit and Linux 64-bit. You can download these files directly from the Admin Portal. This download version was tested with Pipeline Pilot 9.5, MongoDB prototype components, and BIOVIA® Insight. Other versions might also work, but were not tested.

| Operating System | RAM | Paging File Size | Disk Free Space |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| <ul style="list-style-type: none">■ Windows: Microsoft® Windows Server 2008 R2, Windows 7 (64-bit only)■ Linux: Red Hat® Enterprise Linux®, CentOS, Fedora (64-bit only). | Minimum 4GB, (8GB is preferred) | Windows: Minimum of 6 - 10GB. For details on how to check and change the paging file size, see https://docs.microsoft.com/en-us/windows/client-management/determine-appropriate-page-file-size . | Depending on MongoDB usage, 100GB minimum |

Notes:

- For information on additional hardware requirements, go to <https://docs.mongodb.com/manual/administration/production-notes/#ProductionNotes-WhatHardware?>.
- BIOVIA Insight uses MongoDB components. Support and configuration information is available in the *BIOVIA Insight Installation Guide*.

Installing and Deploying MongoDB

You can configure MongoDB to work with Pipeline Pilot in the following ways:

- **Enterprise and medium size deployments:** For enterprise and medium size deployments with over five simultaneous users, MongoDB must be installed on a separate server from Pipeline Pilot. This is due to the fact that MongoDB uses large amounts of RAM, thus competing for system resources with other Pipeline Pilot processes.
- **Small or single-user deployment:** For small deployments with 1-5 users on servers with sufficient RAM (8GB or more) MongoDB can be installed on the same server as Pipeline Pilot. MongoDB binaries are located in <pps_install>/apps/scitegic/mongodb/bin/<OS_VERSION>/bin.

Tip: See <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/>.

Installing MongoDB on a Remote Windows Server

To download MongoDB to a remote Windows server:

1. Connect to the Admin Portal from a browser on the target server.
2. Go to **Maintenance > Download MongoDB**. The Download MongoDB page opens.
3. Select **Windows 64**.
4. Click **Download**.
5. Extract to C:\mongodb.
6. Rename "C:\mongodb\mongodb.config.sample" to "C:\mongodb\mongodb.config".
7. Open "C:\mongodb\mongodb.config" and change the following:

```
dbpath = C:\mongodb\data
logpath = C:\mongodb\logs\mongodb.log
```
8. Save "mongodb.config".
9. Create "C:\mongodb\logs" folder.
10. Create "C:\mongodb\data" folder.

To install MongoDB as a Windows service:

- Open a command prompt and run the following command as administrator:

```
<path to MongoDB bin folder>\mongod.exe --config
C:\mongodb\mongodb.config --install
```

To configure the MongoDB service to start automatically at startup:

1. Open the Services Control Panel (**Start > Administrative Tools > Services**).
2. Double-click the **MongoDB** service.
3. Set the Startup type to **Automatic**.

4. Click **OK**.
5. To start the MongoDB service, click **Start**.

To stop and remove the MongoDB service:

- Open a command prompt and run the following command as administrator:

```
<path to MongoDB bin folder>\mongod.exe --remove --serviceName "MongoDB"
```

Installing MongoDB on a Pipeline Pilot Windows Server

To install MongoDB as a Windows service:

- Open a command prompt and run the following command as administrator:

```
<pps_install>/bin/install_mongo_service.bat "MongoDB"
```

To configure the MongoDB service to start automatically at startup:

1. Open the Services Control Panel (**Start > Administrative Tools > Services**).
2. Double-click the **MongoDB** service.
3. Set the **Startup type** to "Automatic".
4. Click **OK**.
5. To start the MongoDB service, click **Start**.

To stop and remove the MongoDB service:

- Open a command prompt and run the following command as administrator:

```
<pps_install>/apps/scitegic/mongodb/bin/win64/bin/mongod.exe --remove --serviceName "MongoDB"
```

Installing MongoDB on a Remote Linux Server

To download MongoDB on a remote Linux server for prototyping:

1. Connect to the Admin Portal from a browser on the target server.
2. Go to **Maintenance > Download MongoDB**.
3. Select **Linux 64**.
4. Click **Download**.
5. Follow the instructions on the MongoDB site:

<https://docs.mongodb.com/v2.6/tutorial/install-mongodb-on-linux/>

To download MongoDB on a remote Linux server for production:

For production use on Red Hat, we recommend using a .rpm package that includes the control scripts to start and stop MongoDB as a daemon process. See:

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-red-hat/>

If you prototyped the use of MongoDB with a manual installation and changed the default configuration file, ensure that the production server is configured to access that file also.

Configuring a Pipeline Pilot Server to use a MongoDB Server

To configure Pipeline Pilot to use the default MongoDB server:

1. In the Admin Portal, go to **Setup > Global Properties**. The Global Properties page opens.
2. In Package, select **BIOVIA/MongoDB**.
3. For Property, click **DefaultConnection**.

4. In **Value**, type the name of your MongoDB server, followed by a colon and the port number if you have configured MongoDB to listen on a non-standard port (e.g., "localhost", "localhost:4444").
5. Click **Update**.
6. Restart the Pipeline Pilot Server.

Configuring MongoDB as a Secure Data Source

If you want to control access to your MongoDB server, you can configure it as a secure Data Source in the Admin Portal.

To configure MongoDB as a secure data source:

1. In the Admin Portal, go to **Setup > Data Sources**. The Data Sources page opens.
2. Click **Add Data Source**.
3. In the Add Data Source form, type the following required information:

| Option | Description |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name | What you want to call your MongoDB data source. This name is then used to identify this data source in the <i>Connection</i> parameters for MongoDB components. |
| Description | A little more detail about your MongoDB data source. |
| Type | Select MongoDB from the dropdown. |
| Access Privileges | Configure Users/Groups who can access this MongoDB server. |
| Server | Name of the server where MongoDB is installed. |
| Port | Port number for your installed MongoDB server (leave it empty if you did not specify any port during MongoDB installation). |
| Optional DB Username/Password | Specify a username and a password that can be used to connect to the MongoDB server. This encrypted information is stored with your Pipeline Pilot installation. |

4. Click **Save**.
5. Start MongoDB shell:


```
C:\mongodb\mongodb-win32-x86_64-2.6.6\bin\mongo.exe
```
6. Type the following in the shell window (use the username and password entered when configuring MongoDB as a secure Data Source):


```
use admin
db.createUser("<DB_USERNAME>", "<PASSWORD>");
```
7. Stop the MongoDB service.
8. Open "C:\mongodb\mongodb.config" and add the following line:


```
auth = true
```
9. Save "mongodb.conf".
10. Start the MongoDB service. It should be running as a secure data source at this point.

To configure Pipeline Pilot to use the secure MongoDB Data Source by default:

1. In the Admin Portal, go to **Setup > Global Properties**. The Global Properties page opens.
2. In Package, select **BIOVIA/MongoDB**.
3. For Property, click **DefaultConnection**.
4. In **Value**, type Name=<MONGODB_DATASOURCE_NAME>.
5. Click **Update**.

Database Schema Update

A number of packages distributed with Pipeline Pilot require configuration of a database schema. You can use the Database Configuration table to perform this task. It allows you to check that a database schema is up-to-date, and if it is not, you can generate a SQL script that you can use to manually update the database schema.

The table lists installed Packages requiring databases and their associated Data Sources.

To validate a schema:

1. From the Admin Portal, go to **Setup > Data Sources** and create a data source that points to the correct database schema.
2. From the Admin Portal, go to **Maintenance > Database Schema Update** and select the **Data Source** for the appropriate package. The status should indicate *Initial*.
3. Click **Validate** to validate the status of the schema.
4. If the schema has not been configured, the status will indicate *Needs Update*. If it has been configured already, the Status will indicate *Valid*.
5. If it needs to be updated, click the **Generate SQL** button to generate the SQL script that you can use to manually update the database schema.

To change the Data Source for an application:

1. Click the **Data Source** and choose a new one from the list.

Importing/Exporting Configurations

You can export and import your server configurations to allow replication of your environment and to back up your work.

This is especially useful in load balanced environments where it is necessary to configure a number of identical servers. Importing the same configuration file across multiple servers can save time and reduce errors. It is also useful when moving your test environment to a production server (or vice versa).

Notes:

- Export/Import is currently only supported when importing configurations on the same operating systems (i.e., no Windows to Linux or vice versa).
- The Import process removes all existing settings on the target server and replaces them with the settings in the imported file. A backup of the current configuration is saved in the public/backup folder. You can use this backup to restore a legacy configuration if necessary.
- The export file format is XML. You can use a text editor to manually tweak your configuration if necessary. Some sections of the XML file related to security contain encrypted information. Although this data cannot be modified, you can remove any sections you do not need.

Exporting a Server Configuration

To export a Pipeline Pilot Server configuration file:

1. Log into the Admin Portal on the server that uses the configuration you want to export.
2. Go to **Maintenance > Export Import Configuration**.
3. Click **Export Configuration**.
4. At the prompt, specify where you want to save the XML file so you can easily download it to another server. The exported XML file is called "Server-config.xmlx" by default.

Importing a Server Configuration

To import a Pipeline Pilot Server configuration file on another server:

1. Log into the Admin Portal on the target server where you want to replicate the server configuration (Windows-to-Windows or Linux-to-Linux).
2. Go to **Maintenance > Export Import Configuration**.
3. Click **Import Configuration**. A dialog opens.
4. Browse to the location where "Server-config.xmlx" is available.
5. Click **Upload**.
6. Restart your Pipeline Pilot Server so the new configuration can take effect.
7. Repeat on each server you want to replicate.

Managing Licenses

Use the License page for viewing license information, and adding and deleting licenses for Pipeline Pilot features and applications.

Adding a license file

1. Click **Add License** in the **License Files** table.
2. Browse to the license file.
3. Click **Upload**.

If you try to upload a license file with the same name as an existing license file, the **Overwrite Duplicates** checkbox will cause the existing file to be overwritten. If unchecked, an error is generated.

Removing a license file

Click the red X next to the license file's name and confirm the deletion.

IMPORTANT! Deleting a license file may result in a non-functioning Pipeline Pilot server!

Viewing license information

The Details table displays all license features and applications defined on your system. You can sort by the Feature Name, Expiration Date, and License File columns. Hover over the column headers to display the filter icon to enable filtering tools.

Managing the Server

Managing Servers

Use the Manage Server page to do the following tasks:

- [Restart a server](#)
- [Configure Java Servers](#)
- [Take a server offline for maintenance](#)
- [Manage Jupyter Notebook Server](#)

Restarting the Server

Several Admin Portal features require you to restart the server for the settings to take effect. You can do this directly in the Admin Portal to save time.

To restart the server:

1. Confirm that no jobs are running on the same server (**Status > Running Jobs**).
2. Select **Maintenance > Manage Server**.
3. Click **Restart Apache**.

Configuring Java Servers

Overview

Pipeline Pilot includes an embedded Apache Tomcat server (<http://tomcat.apache.org/>). Tomcat must be installed and running to support core services such as task scheduling, fast chart updates, and Query Service. The Pipeline Pilot installer automatically configures and starts a Tomcat service for you.

Notes:

- Apache web server acts as a front-end filter to all Tomcat requests. Apache validates the security of all requests through the security model of the Pipeline Pilot Server.
- Pipeline Pilot is tested and tuned to work with specific applications and services. We recommend that you do not deploy custom or third-party applications to the Tomcat server that is embedded on the platform.

Java Server Configuration

Tomcat uses up to 512 MB of heap memory by default. For larger deployments, you might need to increase Tomcat's maximum JVM heap size to achieve better performance and stability.

To increase Tomcat's maximum JVM heap memory to 1 GB:

1. Go to **Maintenance > Manage Server**.
2. In JVM Options, replace "-Xmx512m" with "-Xmx1024m".
3. Click **Save** to save the JVM Options.
4. Click **Restart Tomcat** so that your changes take effect.

You can configure other JVM start-up options in a similar fashion, including adding new properties as required by installed applications. Pipeline Pilot requires the following settings:

| JVM Option | Purpose | Recommendation |
|-----------------------------------------------------------------------------|--------------------------------------------------------------|-------------------------------------------|
| -Xms128m | Defines the minimum JVM heap memory size. | Required. Not less than 128m |
| -Xmx512m | Defines the maximum JVM heap memory size. | Required. Not less than 512m |
| -Xss512k | Defines the JVM thread stack size. | Optional. Not less than 512k |
| -XX:MaxPermSize | Defines the memory allocated to the permanent generation. | Required. Not less than 128m |
| Dorg.apache.catalina.loader.webappClassLoader.ENABLE_CLEAR_REFERENCES=false | Addresses a bug with Log4J logging from Tomcat. | Required on Windows servers. Do not edit. |
| -Dfile.encoding=UTF8 | Sets the default character encoding to UTF-8. | Required on Windows servers. Do not edit. |
| -Djava.awt.headless=true | Sets AWT to use headless mode on servers with no video card. | Required on Linux servers. Do not edit. |
| -Dlog4j.configuration=log4j.properties | Sets the Log4J logging properties file. | Required on Linux servers. Do not edit. |

Notes:

- Pipeline Pilot is tested and tuned to work with specific application and services deployed by Tomcat. We recommend that you do not deploy custom or third-party applications to the Tomcat server embedded in the platform.
- For further details, see Oracle's Java Standard Edition documentation on JVM options: <https://www.oracle.com/java/technologies/javase/vmoptions-jsp.html>

Taking the Server Offline for Maintenance

The Admin Portal includes a Maintenance Mode feature that can stop new jobs from being launched while allowing existing jobs to complete. This allows administrators on busy systems to gradually shut down a server for maintenance.

To put a server into maintenance mode:

1. Go to **Maintenance > Manage Server**.
2. In the Maintenance Mode section (bottom of page), check **Enable Maintenance Mode**.
3. (Optional) You can specify a user-friendly message to let your users know why the server is temporarily unavailable. Enter your text in **Message**.
4. Click **Save**.

To take your server out of maintenance mode and put it back online:

1. Go to **Maintenance > Manage Server**.
2. In the Maintenance Mode section, uncheck **Enable Maintenance Mode**.
3. Click **Save**.

Tips:

- When the server is in maintenance mode, web service job requests will return an HTTP 503 error. If you have a load balancer that has the capacity to check servers through web requests, the load balancer can detect maintenance mode by issuing a GET request to:
`http://server:port/scitegic/serverstatus?format=json&maint503=1`
- If the server is in maintenance mode, the above request will return an HTTP 503 error code.

Managing the Jupyter Notebook Server

- Jupyter Notebook runs on a separate Jupyter server that is installed with Pipeline Pilot.
- Jupyter Notebook is installed with the `scitegic/jupyter` package that comes with Pipeline Pilot. You will also need Python, which is included in the `scitegic/integration` package.
- Always start or stop the Jupyter service from **Pipeline Pilot Admin Portal > Maintenance > Manage Server > Manage Jupyter Notebook Server**.

On Windows:

- When you **start** the service, in the Windows Services panel, ensure that the “BIOVIA Pipeline Pilot 2021 Service (Jupyter Notebook)” startup type is set to **Automatic** instead of Manual so that the service will restart following a Windows reboot.
- When you **stop** the service, in the Windows Services panel, ensure that the “BIOVIA Pipeline Pilot 2021 Service (Jupyter Notebook)” startup type is set to **Manual** instead of Automatic to prevent the service restarting following a Windows reboot.
- Jupyter Notebook uses the **Setup > Folder Locations > Jobs Directory** defined in **Pipeline Pilot Admin Portal**. If you change it, you must restart the Jupyter service.
- The *Python Jupyter Notebook (on Server)* component is *not* supported in parallel subprotocols or with load-balanced or reverse proxy Pipeline Pilot server configurations.

Managing the XMLDB

Backing Up XMLDB Files

The protocol database (XMLDB) is where all client-accessible protocols and components reside on your server. Whenever these XML files are accessed by the server, they are handled in some area of the XMLDB.

Assuming Pipeline Pilot is installed in `C:\Program Files\BIOVIA\PPS`, the XMLDB consists of the following:

| Data Location | Description |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| \XMLDB\Components | Database of all components, including separately installed packages, and any customized files published by users to the shared tab. |
| \XMLDB\Protocols | Database of all protocols, including separately installed packages, and any customized files published by users to the shared tab. |

| Data Location | Description |
|------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| \XMLDB\Users\ UserNameA UserNameB UserNameC, etc. | Database of all protocols created by users and saved to their individual "User Name" tabs. |

Backup guidelines

It is good practice to frequently back up the XMLDB on your server. This database is continually modified by users as they update their components and protocols. How often you perform the backup depends on how much data you are willing to lose in case of a disk failure or other catastrophe. In such an event, any new data saved on the server since your last backup is lost.

Tips:

- We recommend performing backups on a weekly (or daily) basis. If you have a large group of users, daily backups would be more beneficial, since the demands on the server are greater with heavy client traffic.
- You might still want to save the XMLDB on a tape storage device just like you do with other important data. You can use any backup method or utility of your choosing for the hard storage backup.
- You might also want to periodically purge your XMLDB to keep it running efficiently. For details, go to [Purging Files in an XMLDB](#).

Backing up your XMLDB

You can create a backup copy of the XMLDB by compressing a copy of the entire XMLDB folder into a different folder.

To back up the XMLDB:

1. Go to **Maintenance > Server Backup**.
2. In **Backup file name**, specify the name you want to assign to your backup. A default name for your backup file is pre-assigned based on the current date. All backup files need to use the extension ".xml db.zip". (Platform-specific file naming conventions apply.)
3. When you are ready to save the backup file, click **Create Backup**.

Tip: Depending on the size of your XMLDB, it may take a few seconds or several minutes to create the backup file.

To configure where backup files are stored:

1. Go to **Maintenance > Server Backup**.
2. In **Backup Folder**, specify the new path.
3. Click **Set**.

Purging XMLDB Files

The XMLDB can grow quite large in size over time, increasing the amount of information stored on disk. To improve server performance, it is a good practice to periodically purge the XMLDB. Purging is handled on a per-tab basis - you can purge individual "User Name" tabs, all user tabs simultaneously, and any of the shared tabs.

How purging impacts versioning

Each time a user saves a component or protocol, the XMLDB saves it as a version. The most current version includes the most current changes. Previous changes are saved as "older versions" and they are numbered consecutively by last saved date. Users always work with the most current version.

When you purge the XMLDB, all versions are deleted except for the most current one. You can also specify how many versions to keep in the XMLDB when you purge.

Tip: Users can retrieve earlier versions of their protocols and components. We recommend that you check with them before purging, in case anyone needs to access older versions. (Users can perform version maintenance in the client to keep specific versions.)

To purge the XMLDB:

1. Go to **Maintenance > Server Clean Up**.
2. In **Select Purge Source**, select what you need to purge:
 - **All:** Purge all users and all shared tabs.
 - **Protocols:** Purge the XMLDB for the Protocols tab.
 - **Components:** Purge the XMLDB for the Components tab.
 - **Users:** Purge all "User Name" tabs or open the Users folder and select an individual tab.
3. In **Purge Method**, select how you want to purge the data:
 - **To purge based on versions:** Select **Purge by Version** and then specify the number of versions to keep. To keep only the latest version, select "1".
 - **To purge based on date:** Select **Purge by Date** and then specify the date in Day/Month/Year format. All files dated on or before the specified date are purged.
4. Click **Purge**. Depending on the amount of data to purge, it may take a several seconds or a few minutes.

Tip: You can also compress the XMLDB to save on disk space. For further information, see [Compressing an XMLDB](#).

Restoring XMLDB Backups

If you used the Admin Portal to create backups of your XMLDB, you can quickly and easily restore files from a backup copy. How it works:

- Backup files are stored in a folder specified in your Admin Portal Settings page and use the extension ".xml db.zip".
- The backup is a compressed copy of the entire XMLDB folder. Restoring converts all files into the XML file formats saved on the server for your components and protocols.
- Assuming Pipeline Pilot is installed in C:\Program Files\BIOVIA\PPS, the files are restored to the subfolder hierarchies within the \XMLDB folder.
- You can only restore the entire backup file, not any specific XML files contained in it. Client users can [recover a version](#) of a protocol or component file.

XMLDB versions

Each time a client saves a protocol or component, the server saves a **version** of it in the XMLDB. The version is a saved copy of the file. The most current version includes the most recently saved information. All versions are numbered consecutively by the last saved date. Clients always work with

the most current version unless they manually elect to open an earlier version (explained in more detail below).

When you restore a backup of the XMLDB, versions newer than the files contained in the backup file are overwritten with the ones that were included in the backup file at the time it was created. Clients may lose the changes made to their files since the last backup file was created on the server. The version numbers will remain the same, but the content in the files will change.

When you restore XML files from a backup copy, all changes to protocols and components since your last backup are overwritten. Use the backup that contains the latest versions to minimize how much rework clients have to do.

Restoring from a backup copy

To restore the XMLDB from a backup copy:

1. Go to **Maintenance > Server Restore**.
2. In **Select Backup Source**, select the name of the backup file you want to use to restore your XMLDB. (All available backups are listed in the dropdown.)
3. To restore a backup from another location, click **Browse...** for **Change Backup Source Location** and use the Open dialog to select the backup file based on a path of your choosing.
4. By default, a copy of your current XMLDB is created before the restore procedure takes place, in case you want to refer to this database for some reason later on. If you do not want a copy of the current XMLDB created, click **Delete existing XMLDB instead of moving**.
5. Click **Restore Backup**. The XMLDB is restored from the backup file. Depending on the amount of data to restore, it may take a few minutes.

Recovering a version of an XML file

To recover a version of a component or protocol:

1. If a client user wants to restore a specific version of a component or protocol, it can be restored manually in the client without the need for a system administrator to restore from an XMLDB backup.
2. Log into the client on the user's machine.
3. From any tab in the Explorer window (Components, Protocols, "User Name"), select the item for which you want to access an earlier version.
4. Right-click and select **Versions**. The Version dialog opens.
5. To open an earlier version, select the version number from the list and click **Open Selected**.
6. Save this file so it is the latest version in the XMLDB.

Compressing an XMLDB

Protocols and components are saved in XML format, which requires a substantial amount of disk space. To reduce the space these files occupy on your server, you can set the XMLDB to save these files in compressed format. When XMLDB compression is enabled, new files that users save in the XMLDB, as well as new versions of existing files, are automatically saved in compressed format. Compression is the default format for the XMLDB.

To automatically compress files:

1. Go to **Setup > Server Configuration**. The Server Configuration page opens.
2. For **Compress XMLDB**, select "Yes". Compressed files are saved with the extension ".xml.gz".

3. Click **Save**.

IMPORTANT! Files saved in the XMLDB prior to enabling the compression feature are not automatically compressed. A utility is available that you can run from the command line to manually compress and uncompress XML files.

To manually compress XML files:

1. Shut down the Apache server.
2. Open a command prompt and change to the appropriate directory:
 - **Windows:** <install_root>/bin
 - **Linux:** <install_root>/linux_bin
3. Run the appropriate command-line utility to compress or uncompress the files (using the prevailing compression setting from the Admin Portal):

To compress all files in the XMLDB to the current compression setting:

```
Dbutil -defaultCompress
```

To compress files for a single tab (Protocols, Components, Users) or a specific "User Name" tab:

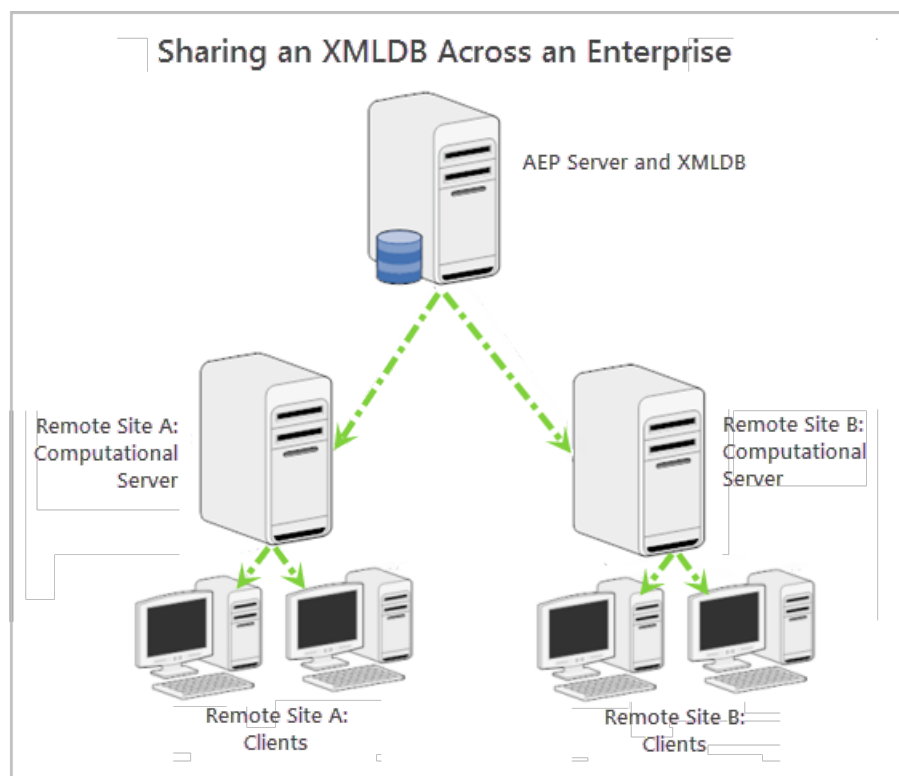
```
Dbutil -defaultCompress [tabname]
```

Sharing an XMLDB with Multiple Servers

A Pipeline Pilot compute server can connect with an XMLDB that is maintained on a separate server. This arrangement allows multiple servers to share a single XMLDB, so clients can share protocols across multiple sites. The computational servers can be situated at any number of sites, and each server can be configured to share a single XMLDB.

To provide this functionality, the computational servers that process protocol jobs at each site need to communicate with one server that hosts the XMLDB for all other servers.

With a shared XMLDB configuration, clients are only aware of the computational server where they run protocols. They have no information about the machine hosting the XMLDB.



Note: The default for a standalone server installation is to use the XMLDB located on that local server.

To configure an XMLDB as a host for sharing:

1. Go to **Setup > Server Configuration**.
2. In **XMLDB Endpoint**, enter the SOAP endpoint for the XMLDB hosting server. (For a clustered deployment, reference the XMLDB service on the primary node of the cluster.)
3. Click **Save**.
4. [Restart the server](#).
5. Repeat on each Pipeline Pilot server that should share this same XMLDB.

Tip: You can log onto each server remotely and perform this task. By default, each server is configured to permit remote logon (Setup > Server > Allow Remote Administration).

If you need to move an installation, do not manually move or delete any files in the XMLDB directory. For further details, see "Moving/Migrating an Installation" in the *Pipeline Pilot Server Installation Guide*.

Chapter 7:

Pipeline Pilot Services

Managing Pipeline Pilot Services

The following services are installed with Pipeline Pilot on Windows:

| Service | Description | Service Type |
|--------------------------------------------------|---------------------------------------------------------|--------------|
| BIOVIA Pipeline Pilot <version> | Standard Pipeline Pilot service that runs on the server | Parent |
| BIOVIA Pipeline Pilot <version> Service (Httpd) | Service that runs your Apache web server. | Child |
| BIOVIA Pipeline Pilot <version> Service (Tomcat) | Service that runs the Tomcat Java application server. | Child |

There are various ways you can control these Windows services. For example, you can:

- Stop and start these services from the Windows Services console (**Administrative Tools > Services**).
- Restart the child services in the Admin Portal (**Maintenance > Manage Server**).
- Stop and restart services using the Pipeline Pilot Manager command line.

Using the Windows Services Console to Manage Pipeline Pilot Services

You can use the Microsoft Management Console (MMC) to manage Pipeline Pilot services (e.g., to stop or start a service). When you stop A Pipeline Pilot service, the ports used by Pipeline Pilot, Apache HTTPD and Java/Tomcat are freed up so you can perform tasks such as upgrading a server.

To manage Pipeline Pilot Services on Windows:

1. Go to **Administrative Tools > Services**. The Services Microsoft Management Console (MMC) opens.
2. Use the right-click menu options to perform service-related tasks: start, stop, pause, resume, and disable.

The Pipeline Pilot services you can manage include:

| Service | Description |
|--------------------------------------------------|----------------------------------------------------------|
| BIOVIA Pipeline Pilot <version> (Manager) | Standard Pipeline Pilot service that runs on the server. |
| BIOVIA Pipeline Pilot <version> Service (Httpd) | Service that runs your Apache web server. |
| BIOVIA Pipeline Pilot <version> Service (Tomcat) | Service that runs the Tomcat Java application server. |

To ensure that all ports are released:

- Open a command window and run the following:
`netstat -a`

```

Administrator: Command Prompt
C:\Program Files\Accelrys\AEP>netstat -a
Active Connections

Proto Local Address           Foreign Address         State
TCP    0.0.0.0:135              vm-aep-w8002:0         LISTENING
TCP    0.0.0.0:445              vm-aep-w8002:0         LISTENING
TCP    0.0.0.0:3389             vm-aep-w8002:0         LISTENING
TCP    0.0.0.0:47001            vm-aep-w8002:0         LISTENING
TCP    0.0.0.0:49152            vm-aep-w8002:0         LISTENING
TCP    0.0.0.0:49153            vm-aep-w8002:0         LISTENING
TCP    0.0.0.0:49154            vm-aep-w8002:0         LISTENING
TCP    0.0.0.0:49171            vm-aep-w8002:0         LISTENING
TCP    0.0.0.0:49175            vm-aep-w8002:0         LISTENING
TCP    10.200.7.174:139         vm-aep-w8002:0         LISTENING
TCP    10.200.7.174:3389       seaglass:62567         ESTABLISHED
TCP    127.0.0.1:9946           vm-aep-w8002:65066     TIME_WAIT
TCP    127.0.0.1:65063         vm-aep-w8002:9904      TIME_WAIT
TCP    127.0.0.1:65064         vm-aep-w8002:9904      TIME_WAIT
TCP    127.0.0.1:65065         vm-aep-w8002:9904      TIME_WAIT
TCP    [::]:135                 vm-aep-w8002:0         LISTENING
TCP    [::]:445                 vm-aep-w8002:0         LISTENING
TCP    [::]:3389                vm-aep-w8002:0         LISTENING
TCP    [::]:47001               vm-aep-w8002:0         LISTENING
TCP    [::]:49152               vm-aep-w8002:0         LISTENING
TCP    [::]:49153               vm-aep-w8002:0         LISTENING
TCP    [::]:49154               vm-aep-w8002:0         LISTENING
TCP    [::]:49171               vm-aep-w8002:0         LISTENING
TCP    [::]:49175               vm-aep-w8002:0         LISTENING
UDP    0.0.0.0:123              *:*:                    *:*
UDP    0.0.0.0:5355             *:*:                    *:*
UDP    10.200.7.174:137         *:*:                    *:*
UDP    10.200.7.174:138         *:*:                    *:*
UDP    127.0.0.1:51641         *:*:                    *:*
UDP    127.0.0.1:55286         *:*:                    *:*
UDP    127.0.0.1:62739         *:*:                    *:*
UDP    127.0.0.1:63277         *:*:                    *:*
UDP    [::]:123                 *:*:                    *:*

C:\Program Files\Accelrys\AEP>

```

Note: It can take a few minutes for Apache to release the ports. To refresh the list, retype the command.

Managing Pipeline Pilot Servers on Linux

On Linux, you can manage Pipeline Pilot using the following commands in your <linux_bin> directory:

- startserver
- restartserver
- stopserver

Shutting Down Pipeline Pilot on Linux

To shut down Pipeline Pilot in a Linux server:

1. Change directories to your server 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 also use the following command to check for running HTTPD processes in the directory:
`> ps -ef | grep httpd`

Manually Starting the Linux Server

IMPORTANT! Perform this task only if you did not create a boot script to automatically start Pipeline Pilot when you installed the application (not recommended). For further information about boot scripts, see the *Pipeline Pilot Server Installation Guide*.

To manually start the server:

1. Change the working directory to the program directory for your Pipeline Pilot installation:
`> cd <pps_install>/linux_bin`
2. Start the server:
`> ./startserver`

Pipeline Pilot Manager

Managing Services with Pipeline Pilot

Overview

AEPManger is a command line tool that lets you perform different tasks with your Pipeline Pilot service and child services. You can stop, start, and restart the Pipeline Pilot service to manage all services together. You can also control child services independently.

Notes:

- AEPManger.exe lives in the "\bin" folder in your installation directory.
- AEPManger features are currently limited to the Windows platform.

Pipeline Pilot Service Tasks

Managing the Parent Pipeline Pilot Service

To manage the Pipeline Pilot service:

`AEPManger.exe -k start [-n <id>]`

`AEPManger.exe -k restart [-n <id>]`

`AEPManger.exe -k stop [-n <id>]`

Note: The optional `-n` parameter is only required if your Pipeline Pilot services were installed using a customized identifier (e.g., when two instances of the same server version are installed side-by-side). In all other cases, the Pipeline Pilot service will have a standard name based on the version, and the `-n` parameter can be ignored.

| To do this task: | Run this command: |
|------------------------------------|---------------------------------------|
| Start the Pipeline Pilot service | <code>AEPManger.exe -k start</code> |
| Restart the Pipeline Pilot service | <code>AEPManger.exe -k restart</code> |
| Stop the Pipeline Pilot service | <code>AEPManger.exe -k stop</code> |

Managing Pipeline Pilot Child Services

To manage Pipeline Pilot child services:

`AEPManger.exe -k restart [-n <id>] [-s <service-name> [-g]]`

`AEPManger.exe -k list`

`AEPManger.exe -k status`

| To do this task: | Run this command: |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| Restart the Apache web server. Include <code>-g</code> for a "graceful" restart, which is faster and just reloads configuration data while the server continues to service requests. | <code>AEPManger.exe -k restart -s httpd [-g]</code> |
| Restart the Apache Tomcat server. | <code>AEPManger.exe -k restart -s tomcat</code> |
| View a list of all registered child services. | <code>AEPManger.exe -k list</code> |
| View the status (stopped, running, etc.) of all Pipeline Pilot services. | <code>AEPManger.exe -k status</code> |

Getting User Assistance

To get help on commands:

`aepmanager.exe -h`

Platform Builder Task

Note: The following information is only for reference, if you need to manually uninstall and reinstall the Pipeline Pilot Server for some reason (e.g., to change the service name). For general purposes, these tasks are achieved by running the Pipeline Pilot installer (recommended).

The features described below are available for anyone who builds on Pipeline Pilot.

To manage the Pipeline Pilot service installation:

`AEPManger.exe -k install [-n <id>]`

`AEPManger.exe -k uninstall [-n <id>]`

| To do this task: | Run this command: |
|----------------------------------------------|---------------------------|
| Install the Pipeline Pilot Manager service | <code>-k install</code> |
| Uninstall the Pipeline Pilot Manager service | <code>-k uninstall</code> |

Notes:

- AEPManager.exe lives in the "\bin" folder in your installation directory.
- Default service names use the ServiceNameSuffix defined in "tokens.config". This is based on the three-digit version by default.
- Child services names are currently "httpd" or "tomcat". A more elegant display name is used for The AEPManager service: "BIOVIA Pipeline Pilot x_x_x <Manager)".
- Side-by-side installations for the same version become easier because of the use of "tokens.config".

Version Change Process

Because of the use of the ServiceNameSuffix in "tokens.config", the service names are "sticky". If you prefer to fall into line with the current version, perform the following steps:

| Step: | Details: |
|---------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| 1. Stop AEPManager. | AEPManager.exe -k stop |
| 2. Uninstall AEPManager. | -k uninstall |
| 3. Refresh the Window Services panel to ensure all is gone. | |
| 4. Edit the "install/tokens.config" file to remove the ServiceNameSuffix line. | |
| 5. Run the token replacement operation. Check that the new ServiceNameSuffix is in place in "tokens.config". | Run Dbutil - |
| 6. Install AEPManager. | -k install |
| 7. Refresh the Window Services panel to ensure all looks good with the reinstalled services. Version numbers should be updated. | |
| 8. Start the AEPManager service. | AEPManager.exe -k start |

Running Pipeline Pilot Services in Console Mode

Since support for the 32-bit platform is phased out, there could be cases (particularly with Microsoft Office applications such as Excel), where the only way to get protocols to run is when Apache (HTTPD) is not running as a service. This is due to the fact that there are security constraints with Microsoft Office running under a Windows service account that causes problems opening Excel files.

The workaround is to run the Pipeline Pilot Service in Console mode. Under this scenario, the Tomcat server runs as a service, and both AEPManager and Pipeline Pilot Server run as command-line executables instead of services.

You can run both AEPManager and the Apache server directly from the command line.

Keep the following in mind when running AEPManager commands in Console mode:

- Console-based commands do not mix with service mode commands.
- Tomcat always runs as a service. AEPManager and Apache always run as command line executables.

Notes:

- AEPManager.exe lives in the "\bin" folder in your installation directory.
- AEPManager features are currently limited to the Windows platform.

Console Mode Commands

| To do this task: | Run this command: |
|-----------------------------------|-------------------------------|
| Start the Pipeline Pilot Server | AEPManager.exe -c -k start |
| Restart the Pipeline Pilot Server | -AEPManager.exe -c -k restart |
| Stop the Pipeline Pilot Server | AEPManager.exe -c -k stop |

Appendix A:

Support for Security Issues

Logon Support

The Admin Portal is only accessible to users who can log onto the system with the *Platform/Administration/Logon* permission.

WARNING!

- Do not remove the *Platform/Administration/Logon* permission from the *Platform/Administrators* group.
- Ensure that an administrator belongs to the *Platform/Administrators* group at all times.

To facilitate logon, the administrator must have the correct authentication information. The Security > Authentication page does not permit changing the authentication method in such a way that the current user cannot log on. For instance, if the current administrator is a file user, and the intention is to only support domain logon, it is necessary to create an administration domain account (to associate at least one domain account to the *Platform/Administrators* group), and then log on as that user before disabling File Authentication.

Note: Default *scitegicamin* credentials are: "scitegicadmin" (name) and "scitegic" (password).

Resolving Administrator Lockout Problems

If an administrator is locked out, consider the following actions (in order):

1. Delete `AuthConfig.xml` and log on using *scitegicadmin* credentials.
2. Delete `Authusers.xml` to reinstall the *scitegicadmin* account.
3. Back up `{root}/xml/db/Object` files one-by-one.

Deleting AuthConfig.xml and Logging on using scitegicadmin Credentials

To delete `AuthConfig.xml` and log on using *scitegicadmin* credentials:

1. Log on to the server.
2. Change to the `{root}/xml/db/Objects` folder.
3. Delete the `AuthConfig.xml` file.

It should now be possible to log on to the Admin Portal with File users security (e.g., using default *scitegicadmin* credentials).

4. If you are able to log on, perform standard administration procedures to set up the appropriate authentication and permission assignments.

Deleting Authusers.xml to Reinstall the scitegicadmin Account

If you tried deleting "`AuthConfig.xml`" and logging on using *scitegicadmin* credentials without success, try the following solution:

To delete `Authusers.xml` and reinstall the *scitegicadmin* account:

1. Try to back up/remove the "`AuthUsers.xml`" file. This causes the *scitegicadmin* account to be reinstalled.

2. Try logging on again with *scitegicadmin* credentials.
3. If this is successful, perform standard administration procedures to set up the appropriate authentication and permission assignments.

Backing up {root}/xmldb/Object files One-by-One

If you tried deleting "Authusers.xml" to reinstall the scitegicadmin account without success, either the scitegicadmin user is not a member of *Platform/Administrators*, or that group does not have the *Platform/Administration/Logon* permission. Try the following solution:

To back up {root}/xmldb/Object files one-by-one:

1. Back up/delete the following {root}/xmldb/Objects files one-by-one until successful:
 - **AuthAssignments.2.xml**: This file specifies administrator overrides of permissions to groups.
 - **AuthGroups.2.xml**: This file specifies administrator overrides of group membership.
2. After removing each file, test your changes:
 - a. Stop the all Pipeline Pilot services.
 - b. From the {root} bin folder, run the command:
`pkgutil -r scitegic/generic -batched`
 - c. Restart the Pipeline Pilot services.
3. Unfortunately, once you are successfully able to log on, it may be necessary to reapply all of your administrative security changes.

Recommended Additional Security Settings

In addition to the authentication, permissions, and certificate behavior you can configure the identification of users access your Pipeline Pilot Server from a browser.

| Setting | Admin Portal Location | Default | Details |
|--------------------------------------------------|------------------------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Expiration of Single Sign-on Credentials | Setup > Server Configuration | 1d | The inactivity timeout before credentials expire. Once credentials expire due to inactivity, the user must log in again. Use d for days, h for hours, or m for minutes. If the user logs in again before the period expires, the credentials are renewed and the inactivity clock is reset. |
| Retain Session Cookie beyond Web Browser Session | Setup > Server Configuration | Yes | Set this to "No" if you want to force users to re-login every time they access a web application from a fresh browser, even on the same machine. |
| Session Salt | Setup > Server Configuration | [empty] | To further enhance security you can specify a different password to encrypt the contents of a session. |

Note: Some other settings can also be configured to optimize performance when configuring Pipeline Pilot as a host for web client applications. For further information, see [Settings for Web Client Hosts](#).

Appendix B:

Client-Server Deployment Issues

The following information addresses common issues for supporting clients on different server Pipeline Pilot deployments:

- [Overview of grid job processing](#)
- [Guidelines for grid-friendly protocols](#)
- [Running protocols on a grid](#)
- [Running a complex job on multiple nodes](#)
- [Batch size recommendations for parallel protocol processing](#)
- [Fine-tuning clustering configurations](#)

Overview of Grid Job Processing

After Pipeline Pilot determines that a protocol can run on the grid, the following events take place:

1. The command line for `scisvr` is modified for running on the grid and written to a file named `"queue.sh"` in the `"job/RunInfo"` directory.
2. A file named `"statsEndpoint.txt"` is generated containing the primary server URL, so the job can send stats messages back to the primary server (job progress).
3. The grid engine specific submit script (based on Admin Portal configuration) is called with the following arguments:
 - Path to the grid engine top-level directory
 - Path to the `"queue.sh"` script for the job (command line for `scisvr_wrapper`)
 - Grid engine-friendly version of the job name
 - Queue name to use if specified.
4. The submit script returns the grid engine job ID, which is then saved in a file named `"gridId.txt"` in the `"job/RunInfo"` directory.
5. Job submission calls the `"<xxx>_submit"` script in the `"linux_bin"` directory, where `"<xxx>"` is the grid engine type selected in the Admin Portal.

Tip: Administrators can optimize grid hardware configurations to process jobs on the grid. To alleviate memory load from the head node on the server, consider any available node on the grid as a potential head node for the protocol job. The child node should have the same privileges as the head node.

Guidelines for Grid-friendly Protocols

When configuring protocols (and subprotocols) for a grid-based deployment, protocol designers should refrain from using components that require user interaction during protocol execution. For example, avoid components such as:

- *Display Message for Each Data*
- *Text Prompt*
- *Data Record Tree Viewer*

If the protocol contains a component that requires user interaction, it is handled on a child node, where it will trigger a warning message that cannot be resolved by an end user. The protocol will eventually stop waiting for the message to get resolved and the job will not finish.

If a protocol contains one or more subprotocols and users experience problems getting the job to complete on your grid, ensure that the protocol does not use any components that require user interaction.

Running Protocols on a Grid

When connected to a server that is enabled for grid support, client users can set a *Run On Grid* parameter at the top-level protocol to run the job on the grid.

| Implementation | |
|-------------------------------------------------|------|
| Tempfiles | |
| Protocol Form | ... |
| <input checked="" type="checkbox"/> Run on Grid | True |
| Queue Name | |

Parameters Implementation Web Service

Although setting the *Run On Grid* parameter to "True" makes a protocol eligible to run on a grid, it does not necessarily ensure that the protocol will run on the grid. Situations that might interfere include:

- Some protocols are not intended to run on a grid under any circumstances (e.g., extremely short-running protocols such as blocking web service protocols).
- If a grid engine path is not specified in the Admin Portal, the job will run on the local server instead of the grid.
- If the *Run On Grid* parameter is enabled and the user imports the protocol on a client that is not connected to a server with a grid enabled, the job can only run on the local server.

Note: When *Run On Grid* is set to "True" in the Implementation tab, and the client is connected to a server with a configured grid engine, the entire protocol is outsourced to a child node for execution, to avoid overloading the server. This behavior is similar to using a cluster.


Running a Complex Job on Multiple Nodes

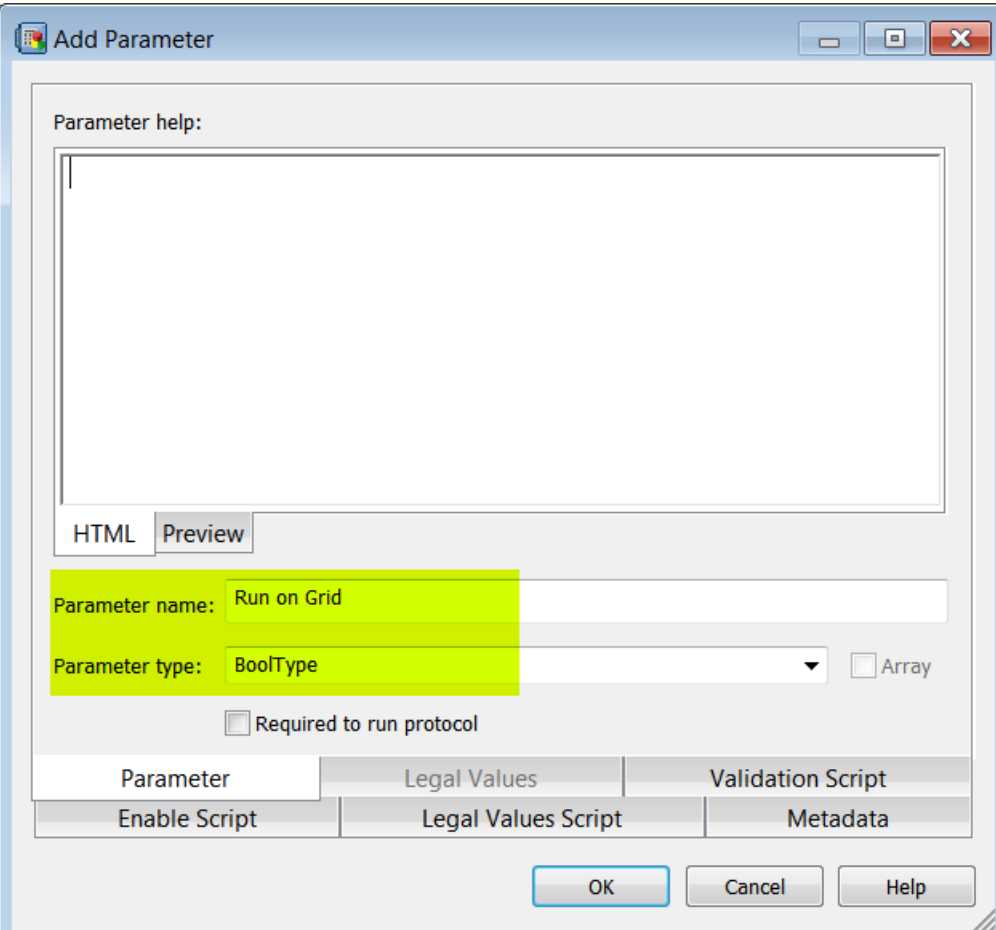
In some instances, a server might not respond after it receives a job that involves multiple calculations or nodes. If this happens, most likely the server's memory resources ran too low to process the job. Workarounds include:

- Upgrade the memory on your server
- Add additional servers via clustering
- Set the *Run On Grid* parameter to "True"

Adding Run on Grid Parameters to Legacy Protocols

The *Run On Grid* parameter can be added to any legacy protocol that needs to run on the grid by doing the following:

1. Open the protocol in Pipeline Pilot.
2. Right-click inside a blank area of the workspace and select **Edit Protocol**.
3. Click **Add Parameter** .
4. In **Parameter name**, type:
Run on Grid
5. For **Parameter type**, select "BoolType".



6. Click **OK** to close the Add Parameter dialog.
7. Click **OK** to close the Edit Protocol dialog.

Batch Size Recommendations for Parallel Protocol Processing

To determine the ideal batch size for a parallel protocol, distinguish between the following scenarios:

Clustering Scenario

You have an instance of Pipeline Pilot running on some or all nodes of your cluster and you need to run a Discovery Studio protocol in parallel. By default, this breaks the input stream in batches of 25 and will present to the available resources for their execution. Discovery Studio bundles all needed information for completion on those nodes.

Grid Scenario

You have a working grid engine associated with your server. The Discovery Studio protocol is aware of this grid and can outsource the calculation to any of the available nodes on the cluster for the default queue, unless you specify a particular queue. You don't need to have an instance of Pipeline Pilot on these nodes, as it is handled by the grid. The grid's head node checks for completions on intervals retrieving the job completed by the child node. You need to consider balance between batch size and intervals retrieving the job. These parameters determine the best batch size for your job. This is an empirical factor that you need to fine-tune for your protocols.

Consider the following:

- Bundle
- Sending
- Calculation
- Collection

If batch is set to "1" and calculations are quick, the overhead is on sending and collection (Grid factors). If batch is set to "1000", your overhead is bundling (Pipeline Pilot /Discovery Studio factor), and you will use only one child node (not the ideal setup).

Depending on your clustering or grid scenario, your solution depends on the type of job you like to run and the parameters of your grid environment.

For example for a data set of 100 records, consider performing the following tests:

- Batches of 1
- Batches of 10
- Batches of 25
- Batches of 100

The results will vary, allowing you to determine the ideal batch size for your needs.

Fine-tuning Clustering Configurations

Enabling Clusters used as Endpoints

If you have multiple Pipeline Pilot Servers, you can enable clustering and point to the servers used as endpoints. This way, the same URL (main server) can be used to access both Pipeline Pilot Servers and servers that run other BIOVIA applications (e.g., Discovery Studio). All child nodes can have an instance of a Pipeline Pilot Server that points to the same location as the main server. The same port combinations are used with multiple servers.

Clustering

Clustered Server Management

(Linux only) You can create a cluster of servers, each with its own instance of the server running from a single NFS-mounted Accelrys Enterprise Platform (AEP) Server installation. Jobs sent to the server will be distributed across the different machines in the cluster. For more information, see the help page or contact Accelrys Technical support. Note that all clustering is disabled if the *Load Balancing* setting below is set to Off.

| | |
|----------------|-----|
| Load Balancing | Off |
|----------------|-----|

Internal Server Name

If you are configuring for clustering or a grid engine, you must specify the internal server name to be used to contact the AEP head node from within the cluster. This allows you to take advantage of local high speed back end networks for clusters and grids. This host name must be visible from inside the network segment for the cluster or grid. You may use an IP address in this field if desired. **This setting is required for clusters and grids.**

| | | |
|----------------------|----------------------|------|
| Internal Server Name | <input type="text"/> | Save |
|----------------------|----------------------|------|

☐ **Private Cluster** (Endpoints are accessible from the primary server, but not from client machines).

Cluster Runner Endpoints

To add a server to a cluster, enter the server name in the text box below and click *Add*. In a cluster, all the endpoint servers run using the same port, since they are all based on a single NFS-mounted installation.

| Active? | Endpoint | Parallel Subprotocols Only | Remove |
|-------------------------------------|----------|---------------------------------------------------------------|-------------------------------------|
| <input checked="" type="checkbox"/> | espresso | <input type="radio"/> Yes <input checked="" type="radio"/> No | <input checked="" type="checkbox"/> |
| <input checked="" type="checkbox"/> | sherry | <input type="radio"/> Yes <input checked="" type="radio"/> No | <input checked="" type="checkbox"/> |
| <input type="text"/> | | Add | |

Notes:

- When jobs are submitted to the main Pipeline Pilot Server, they are distributed based on node availability.
- Additional licenses are required for each Pipeline Pilot Server instance. For additional information, consult your BIOVIA account representative.

Cluster Job Handling

When clustering is deployed, protocol jobs are distributed to different servers by the main server, based on node availability. You can track server/node usage in the Admin Portal (Reports > Completed Jobs).

| Job | Job Version | User | Client | Server | CPUs Used |
|------------------------------------|-------------|-------|----------|----------|-----------|
| Count and Index Data Example | 1 | userA | Aardvark | sherry | 2 |
| Count and Index Data Example | 1 | userA | Aardvark | sherry | 2 |
| Cleanup- Data | 3 | userB | Bumble | drexl | 2 |
| Cleanup Data | 3 | userB | Bumble | drexl | 2 |
| Perform Calculation on all Members | 6 | userC | Caravan | espresso | 2 |
| Perform Calculation on all Members | 6 | userC | Caravan | espresso | 2 |

Job processing on a cluster where node usage is tracked as part of the Completed Jobs information.

Job Folder Maintenance

A "job" is an instance of a running process, plus an organization of folders and files that contain all the data related to that protocol's execution. Once the running process completes, the job folder may remain, and the lifetime of the job folder is determined by which interface was used to launch the job initially.

Job Folder Lifetimes

Here are details on various job folder lifetimes for jobs created from various Pipeline Pilot interfaces:

- **Pipeline Pilot Client:** Pipeline Pilot jobs are created with a jobdir whose name starts with the letters "ppc", for which a set number are kept, depending on a combination of personal settings for number of jobs to save in the client (**Tools > Options > Jobs tab list size**) and the **Maximum Archived Jobs per Pipeline Pilot User** setting in the Admin Portal (**Setup > Job Settings**). Job folders outside these limits are deleted when the client session is closed.

Note: The Admin Portal limit applies only to jobs created in Pipeline Pilot Client.

- **Web Port:** Web Port jobs are created with a jobdir whose name starts "wpt". Job folders are kept until the user explicitly deletes them from the Web Port Jobs tab, allowing users the ability to re-run previously launched jobs. Web Port users should get in the habit of occasionally looking at the Jobs tab and deleting protocols they no longer wish to reuse. (Web Port may also create temporary utility job folders without this job folder naming scheme).
- **SharePoint Bridge:** For jobs invoked via the SharePoint Bridge client, a session cookie is typically stored and used. Session cookies default to expiring after 30 days, and can be customized in the Admin Portal setting "Expiration (days, hours or minutes) of Single Sign-on Credentials" (Setup > Server Configuration). After the session cookie expires, previously run jobs may not be available for viewing from SharePoint. The session cookie is renewed each time a protocol on a SharePoint page is explicitly re-run.
- **Blocking Web Services:** By default blocking jobs are deleted when the job completes. Strictly speaking, they are marked for deletion with an expiry that is either zero, or 10 minutes if the web service includes at least one job folder file in its result. This is so that the web service client has some time to retrieve the job results before it is deleted. Note that there is a distinction between what files a protocol may write out and what a web service wrapper may define as the result. It is the web service definition that is used to judge whether a job can be removed immediately (because its data results have been returned in the service response) or needs to be retained for a short period.

The default expiration behavior can be overridden by web service clients or by web services themselves by extending the expiry to any amount or by having no automatic expiry. In this latter case, the behavior is more like that of a non-blocking service (see below).

Job expiry is mediated by a file named ".expiration" in the "RunInfo" subfolder of the job. The 3rd line of the file indicates in human readable form when the job is set to expire (0 means "Now"). Completed jobs that are expired (i.e., jobs that have expiration files whose date has passed) are eligible for cleanup. The clean up sweep happens very frequently, so expired jobs will not exist for more than a couple of seconds after expiry.

- **Non-Blocking Web Services:** In the case of a non-blocking (or asynchronous) service, the job is launched by the server and then left to run. The client will typically monitor its progress and retrieve results when complete. The client can then delete the job using an appropriate API, depending on the SDK or API they are to run and manage job. Non-blocking jobs can also be run with an expiration behavior, instead of being left for the client to delete.

- **Runprotocol.exe:** Job folders are immediately deleted unless the -GUI flag is used. When the -GUI flag is used, at job process completion, the job folder is deleted after the specified seconds have elapsed. If a value is not specified, you are prompted for a key press to delete the job.
- **SDKs:** Job folders created by Pipeline Pilot SDKs (for example, Java and .NET) are created with no expiry. The Pipeline Pilot developer is expected to call functions such as `pp.ReleaseJob` explicitly to handle the cleanup.
- **Discovery Studio:** For protocols that run via Discovery Studio, the jobs are deleted from the server once the user selects them for download to Discovery Studio.

Because these job folders, particularly for jobs invoked via Web Port, can accumulate over time and take significant storage space, Pipeline Pilot provides the ability to redirect the Jobs folder to a larger drive location, configured in the Admin Portal settings (**Setup > Folder Locations**).

If this is not possible or insufficient, you might institute a policy where job folders older than a set age are removed, and users are warned that this will occur. Deleting older jobs folders prevents users from being able to see the previous job settings, protocols and results. It also prevents those jobs from being readily reopened and re-run from (for example, from the Web Port Jobs tab). This does not impact any protocols or data that the user properly stored in the XMLDB or in their user directories. This type of cleanup could be done via a Pipeline Pilot protocol that runs via a scheduled task or cron to delete defined jobs folders older than a certain date.

Maximum Job Age Based on Job Folder Size

In **Setup > Job Settings**, you can set a **Maximum Job Age Based on Job Folder Size**. This setting controls clean up of users' jobs based on job age and minimum folder size in KB, MB, and GB. You can add multiple rows to define the maximum age for different folder sizes. Jobs that are set to expire at a future date are ignored. Cleanup is scheduled every four hours.