pure::variants Setup Guide

pure-systems GmbH

Version 5.0.3.685 for pure::variants 5.0

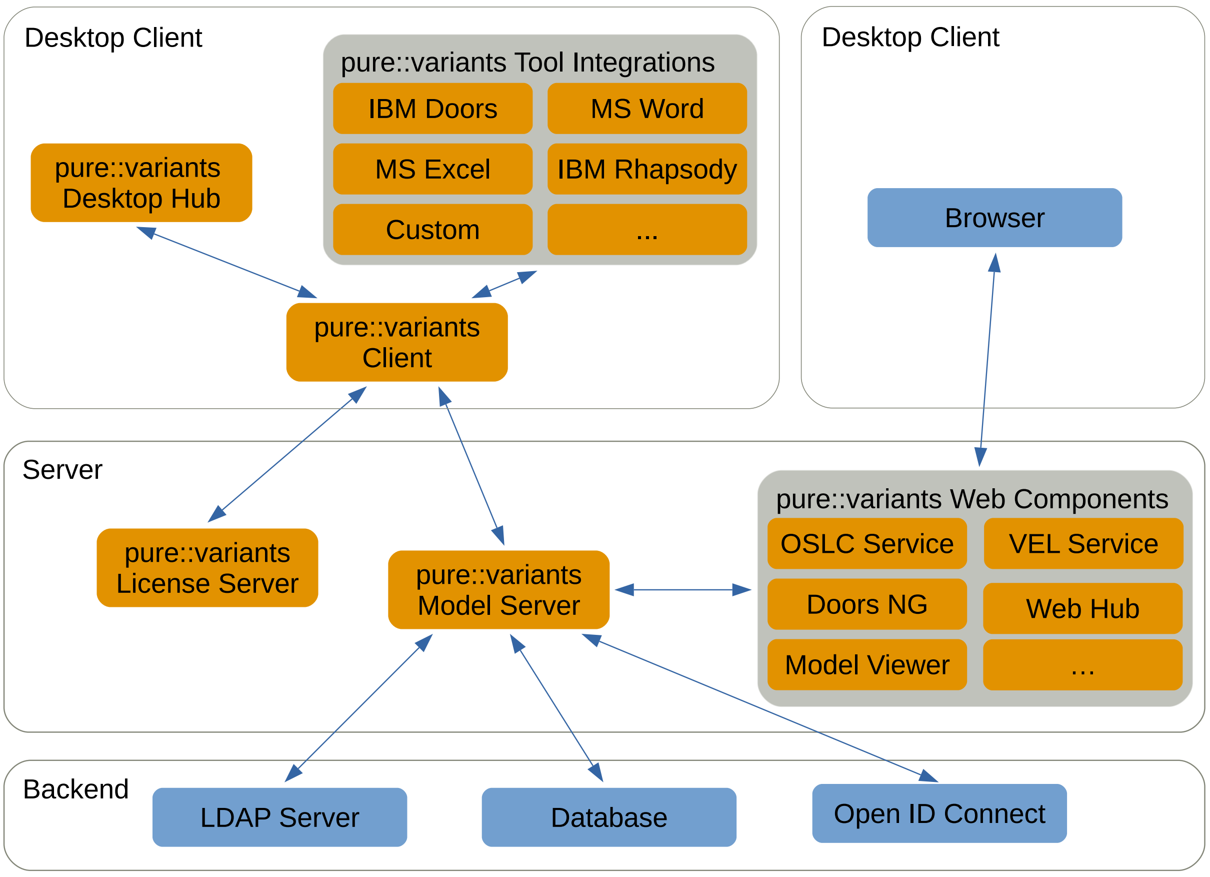
2020

Table of Contents

1. Introduction

Depending on the engineering tool landscape in place and on the desired data management mode for pure::variants models, the pure::variants setup consists of some or all of the components depicted as orange boxes in figure Figure 1, "The Big Picture". In this setup guide all components are described that have to be installed, updated, or configured to cover the different possible deployment scenarios. The components are grouped from IT perspective into components running on a server, and components running on the desktop clients. In one deployment scenario, only desktop client components depicted in the upper left might be needed, while in another scenario server components combined with a browser running on client side might be needed. There are also some boxes visualized in blue, e.g. Browser and LDAP, that are not delivered by pure-systems but interact with pure::variants components in certain deployment scenarios. In the following sections each of the orange boxes are explained from the perspective of administration and setup.

Figure 1. The Big Picture



The manual is available in online help inside the installed product as well as in printable PDF format. Get the PDF here.

2. System Requirements

pure::variants has different system requirements, depending on the part of the software going to be installed. The following table lists the system requirements for all parts of the software.

Table 1. System Requirements

pure::variants Software	OS	Software	Memory
Eclipse based Client	Windows 7, Windows 8, Windows 10	Java SE 8 64-Bit	Minimum: 2 GB Recommended: 6 GB

pure::variants Software	OS	Software	Memory
	Windows Server 2003, 2008, 2012, 2016, 2019 Linux 64 Bit with X11 Window System installed MacOS	Eclipse from 3.8.0 to 4.17	
Model Server with Database	Windows 7, Windows 8, Windows 10 Windows Server 2003, 2008, 2012, 2016, 2019	Oracle 9 or newer MSSQL Server 2008 or newer	Minimum: 2 GB Recommended: 8 GB
License Server	Windows 7, Windows 8, Windows 10 Windows Server 2003, 2008, 2012, 2016, 2019 Linux 64 Bit		Minimum: 512 MB Recommended: 1 GB
Web Components	Windows 7, Windows 8, Windows 10 Windows Server 2003, 2008, 2012, 2016, 2019 Linux 64 Bit	Oracle JDK/JRE Java SE 8 or OpenJDK 8 Apache Tomcat 8 or newer, WebSphere Liberty Kernel v19.0.0.6 or newer	Minimum: 2 GB Recommended: 6 GB

The Java compatibility is tested with the official Java Standard Edition provided by Oracle (https://www.java.com/en/download/) and the OpenJDK provided by Oracle (https://jdk.java.net/archive/).

3. pure::variants Client

pure::variants can be installed using the pure::variants installer as a stand-alone application or it can be installed into an existing Eclipse based tool chain. For both ways to install pure::variants we recommend to use the pure::variants installer.

The pure::variants installer is available for Windows only. If the operating system platform is Linux or MacOS X, pure::variants needs to be installed into an existing Eclipse instance. See Section 3.1.2, “Install into an existing Eclipse”.

In case of very strict firewalls or no network access on the installation machine either install pure::variants as a stand-alone application. (Section 3.1.1, “Install with pure::variants Installer”) or install pure::variants into an existing Eclipse instance using an update site. (the section called “Using update site”). These installation methods allow you to first download the installation packages and install pure::variants afterwards.

The installation procedures are described below. Once the initial installation has finished, installation of a license is required to use pure::variants. See following section for more information on license installation.

3.1. Install pure::variants Client

3.1.1. Install with pure::variants Installer

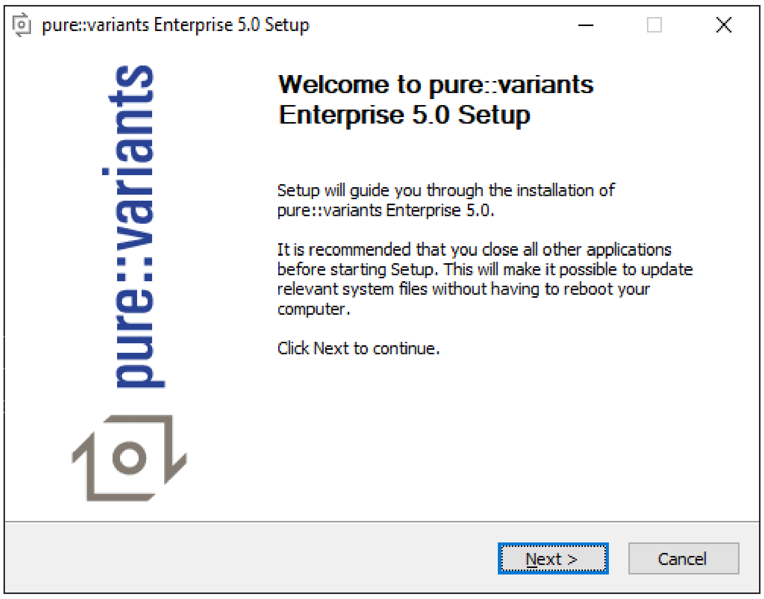
This installation method is available for Windows only. If you do not use Windows please see Section 3.1.2, “Install into an existing Eclipse”.

The Windows Installer can be downloaded from the pure::variants product web page. Go to http://www.pure-systems.com/pvde-update. The product download pages are protected by a password. You need to login by using the email address and the registration number from the license file.

Download the installer package ("pure::variants Windows Installer Package") and extract it. The installer will set up a fresh Eclipse with pure::variants and documentation. Start the installation by double-clicking "Setup Enterprise X.Y.ZZ.exe". Running the pure::variants enterprise installer requires Administrator privileges.

All pure::variants extensions available for the account are automatically included in the Windows Installer download. However, some may not be enabled by default in Installer. Make sure to select the desired extensions during the installation process. Later updates to the extension selection can be done either by reinstalling pure::variants or by following the alternatives described in the section called “Using update site”.

Figure 2. pure::variants Client Installer



Welcome to pure::variants Enterprise 5.0 Setup

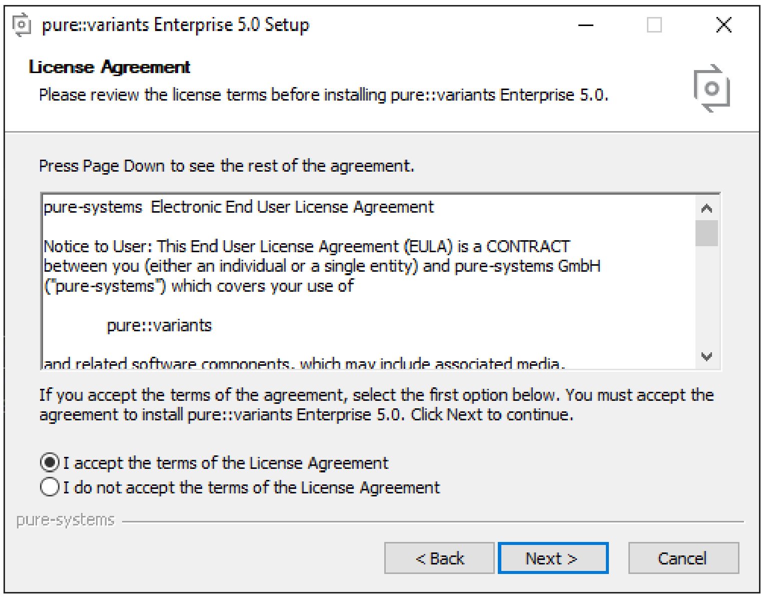
Setup will guide you through the installation of pure::variants Enterprise 5.0.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Click *Next*.

Figure 3. Setup pure::variants Client License



License Agreement

Please review the license terms before installing pure::variants Enterprise 5.0.

Press Page Down to see the rest of the agreement.

pure-systems Electronic End User License Agreement

Notice to User: This End User License Agreement (EULA) is a CONTRACT between you (either an individual or a single entity) and pure-systems GmbH ("pure-systems") which covers your use of

pure::variants

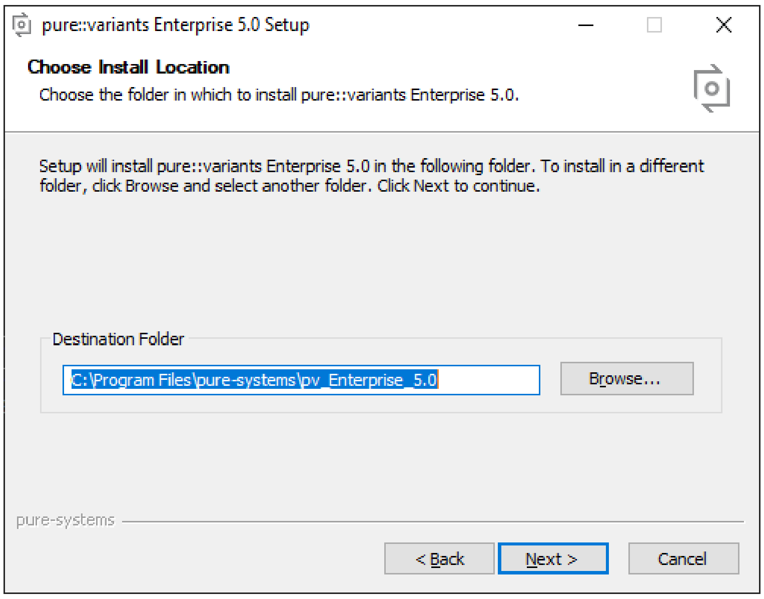
and related software components, which may include associated media,

If you accept the terms of the agreement, select the first option below. You must accept the agreement to install pure::variants Enterprise 5.0. Click Next to continue.

[✓ I accept the terms of the License Agreement]
I do not accept the terms of the License Agreement

Read the license agreement and after accepting it click *Next*.

Figure 4. Setup pure::variants Client Installation Location


pure::variants Enterprise 5.0 Setup

Choose Install Location

Choose the folder in which to install pure::variants Enterprise 5.0.

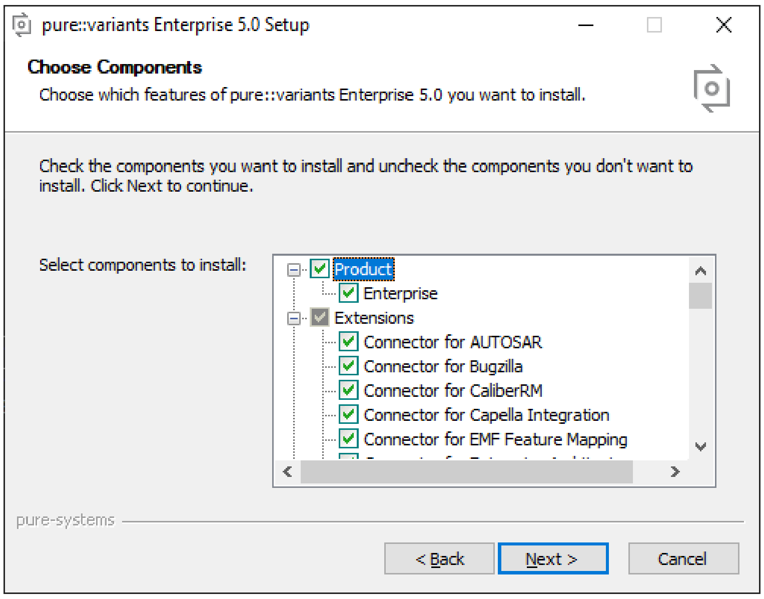
Setup will install pure::variants Enterprise 5.0 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\Program Files\pure-systems\pv_Enterprise_5.0

Select the folder where to install the pure::variants client files. Click *Next*.

Figure 5. pure::variants Client Feature Selection


pure::variants Enterprise 5.0 Setup

Choose Components

Choose which features of pure::variants Enterprise 5.0 you want to install.

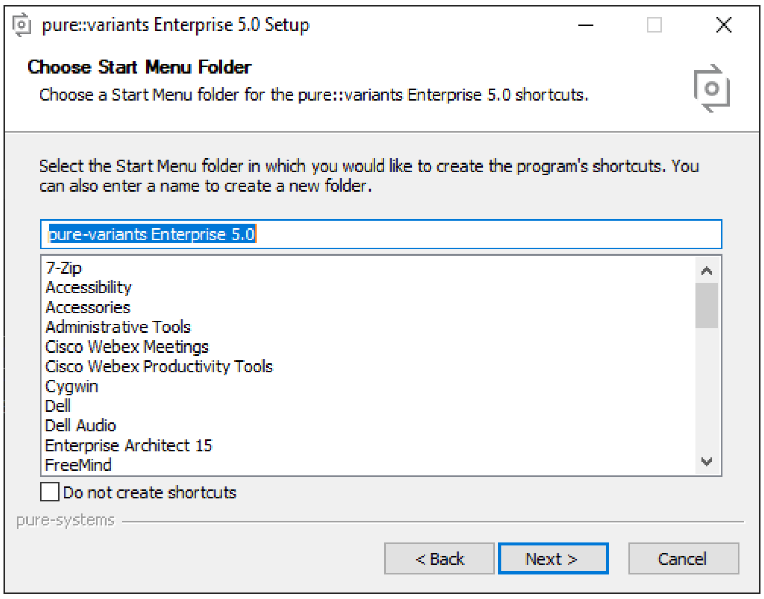
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- ✓ Product
  - ✓ Enterprise
- ✓ Extensions
  - ✓ Connector for AUTOSAR
  - ✓ Connector for Bugzilla
  - ✓ Connector for CaliberRM
  - ✓ Connector for Capella Integration
  - ✓ Connector for EMF Feature Mapping

Select the connectors which shall be installed with the pure::variants client. Click *Next* after the feature selection is complete.

Figure 6. Setup pure::variants Client Start Menu


pure::variants Enterprise 5.0 Setup

Choose Start Menu Folder

Choose a Start Menu folder for the pure::variants Enterprise 5.0 shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

pure-variants Enterprise 5.0

7-Zip
Accessibility
Accessories
Administrative Tools
Cisco Webex Meetings
Cisco Webex Productivity Tools
Cygwin
Dell
Dell Audio
Enterprise Architect 15
FreeMind

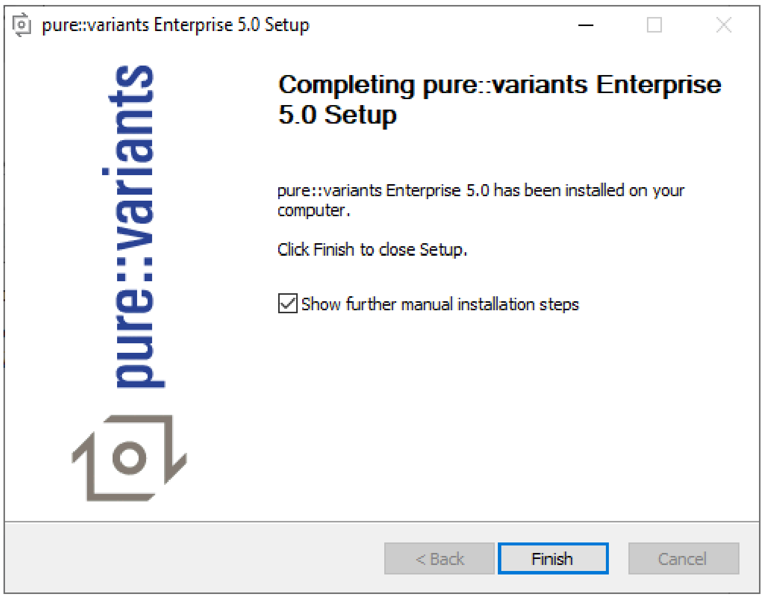
☐ Do not create shortcuts

Enter the name for the Windows start menu entry, or disable the creation of the start menu entry. Click *Next*

The next pages may show information about pure::variants integrations, which are installed along with the pure::variants client. If no connector was selected providing an integration, this page will show the *Install* button.

Click *Install* to start the installation process.

Figure 7. Setup pure::variants Client Finish Page


pure::variants Enterprise 5.0 Setup

Completing pure::variants Enterprise 5.0 Setup

pure::variants Enterprise 5.0 has been installed on your computer.

Click Finish to close Setup.

☑ Show further manual installation steps

The Option *Show further manual installation steps* will open a text document showing more information about the installed integrations and possible manual installation steps, which have to be performed for the integrations to work properly.

pure::variants Enterprise Installer Command Line Options

The pure::variants installer provides the following command line options:

Table 2. pure::variants Installer Command Line Options

Option	Description
/S	Run the installation in silent mode. No installation dialog is opened. Automatically installs the default selected software packages, or all if used together with option /ALL.
/ALL	Select all packages for installation.
/32	Force installation of 32-bit binaries, also on 64-bit operating systems.
/NODOTNET	Skip installation of the .NET 4 Framework.
/NOINTCOMP	Skip installation of the integration components for Java & .NET.
/JAVA	Location of the Java executable to be used for the installation. Example: /JAVA="C:\Program Files\Java\jre6\bin\java.exe"
/ECLIPSE	Path to an existing Eclipse installation into which to install pure::variants as a feature, instead of installing pure::variants as a stand-alone application. This directory must contain the file eclipsec.exe. Example: /ECLIPSE="C:\Program Files\Eclipse 3.8\eclipse"
/D	Path to the directory where to install the pure::variants stand-alone application. Must be the last option on the command line and must not contain any quotes, even if the path contains spaces. Example: /D=C:\Program Files\pure-variants

Install pure::variants in silent mode

The pure::variants Client installer has a silent mode. This mode installs the pure::variants client without user interaction by just using the standard settings of the pure::variants client installer also considering further options on the command line.

To do this, call the installer with command line option */S*. See the section called “pure::variants Enterprise Installer Command Line Options” for all available command line options.

3.1.2. Install into an existing Eclipse

pure::variant can be installed into an existing Eclipse based tool chain. To install pure::variants, the pure::variants installer package downloaded from the pure::variants updatesite can be used. We recommend this for all Windows users.

Alternatively the pure::variants update site can be used directly with the Eclipse client. You can also download an archived update site from the pure::variants update site and use this with the Eclipse client (See the section called “Using update site”).

Prerequisites

pure::variants needs to following features to already be installed in the target Eclipse, or the Eclipse instance has to have access to the Eclipse release update site.

- JavaScript Development Tools
  - org.eclipse.wst.jsdt.feature.feature.group
- Eclipse Business Intelligence and Reporting Tools (BIRT)

- org.eclipse.birt.feature.group

- Graphical Modeling Framework
  - org.eclipse.gmf.feature.group

Using pure::variants Installer

The installation into an existing Eclipse instance is done the same way as installing pure::variants as stand-alone application (See Section 3.1.1, “Install with pure::variants Installer”).

There is one difference: the target Eclipse has to be defined with the */ECLIPSE* command line option.

Using update site

- Start pure::variants (or the Eclipse into which pure::variants has been installed).

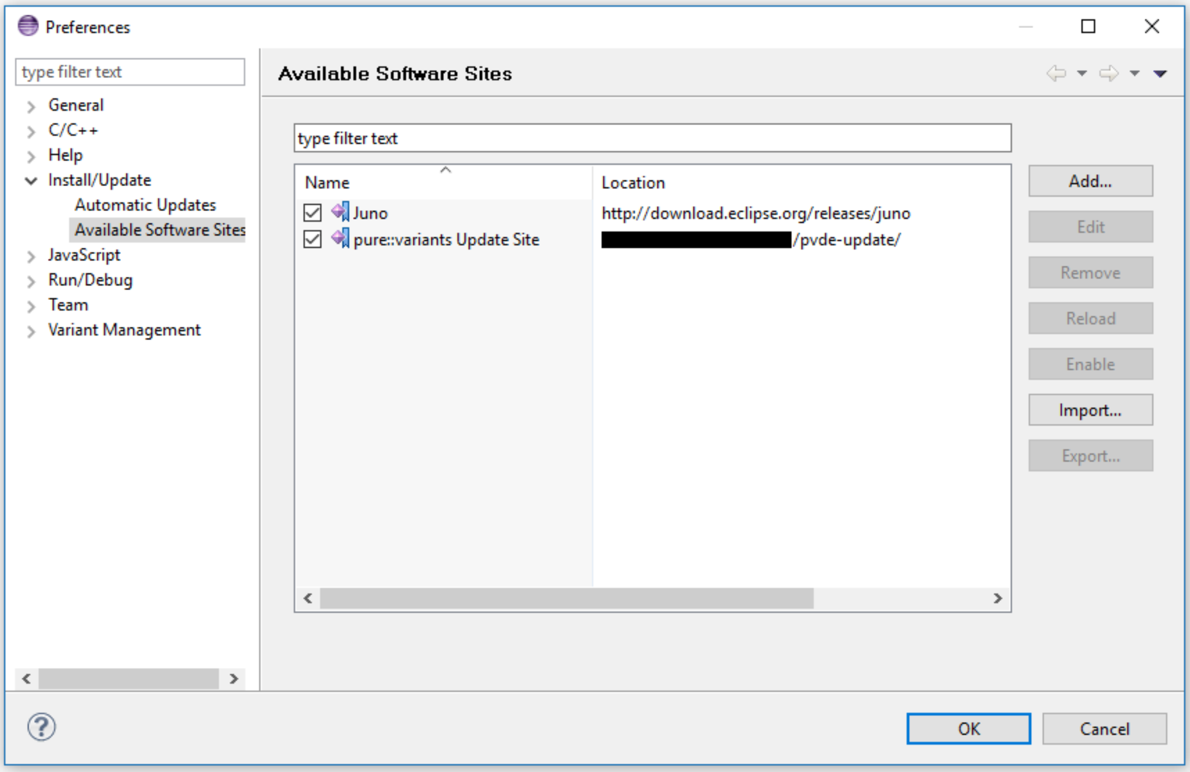
- Select "Help"->"Install New Software...".

- Select "pure::variants update site" from the available Software Sites.

  If location "pure::variants update site" is not present, enter your location in the edit field, or press "Add" if you have a local update site at hand.

  The location of the site depends on the pure::variants product variant. Visit the pure-systems web site (http://www.pure-systems.com/pv) or read your registration email to find out which site is relevant for the version of the software you are using.

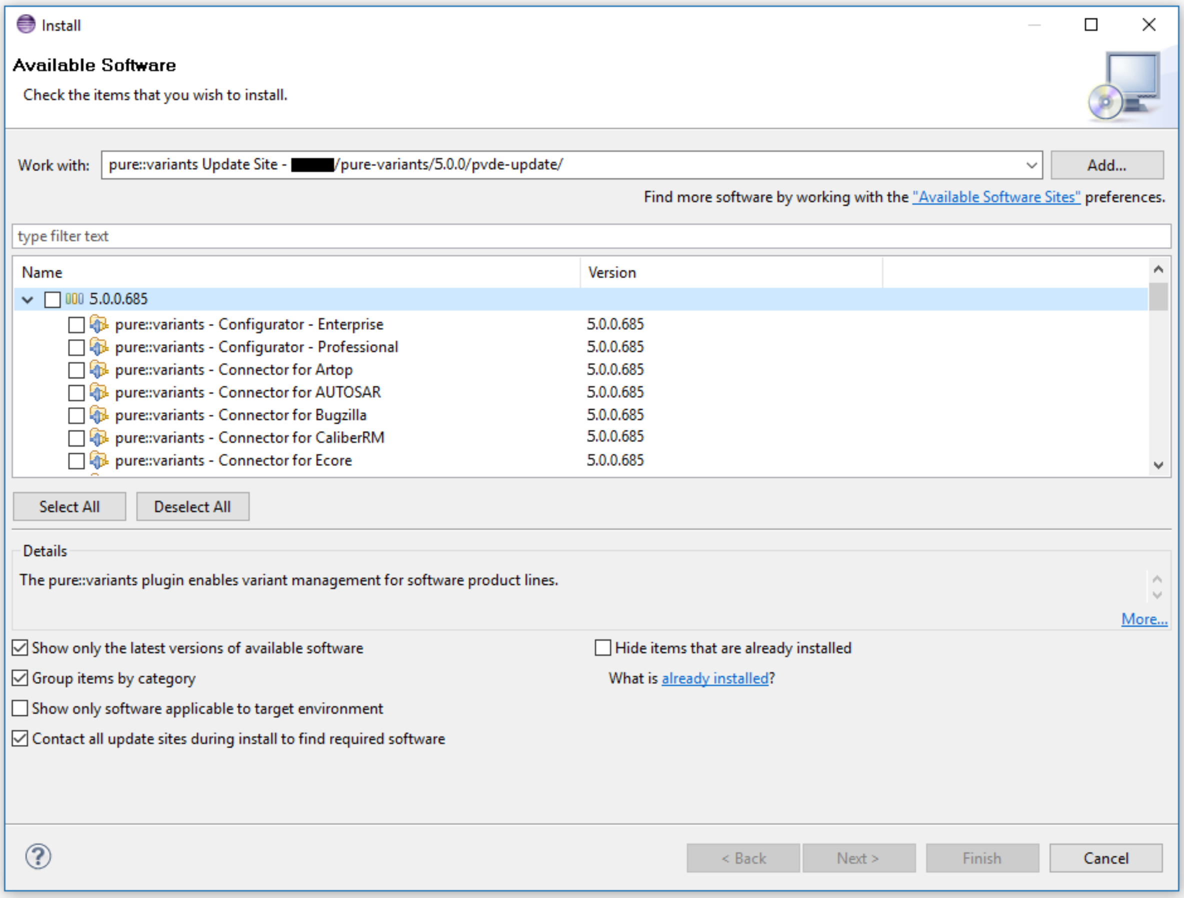
Figure 8. Update Site Selection



Name	Location
☑ Juno	http://download.eclipse.org/releases/juno
☑ pure::variants Update Site	[redacted]/pvde-update/

- Unfold the pure::variants update site and select all features to be updated. Select "Next".

Figure 9. Pure::variants Plugin Selection



Available Software

Check the items that you wish to install.

Work with: pure::variants Update Site - [redacted]/pure-variants/5.0.0/pvde-update/

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☐ 5.0.0.685	
☐ pure::variants - Configurator - Enterprise	5.0.0.685
☐ pure::variants - Configurator - Professional	5.0.0.685
☐ pure::variants - Connector for Artop	5.0.0.685
☐ pure::variants - Connector for AUTOSAR	5.0.0.685
☐ pure::variants - Connector for Bugzilla	5.0.0.685
☐ pure::variants - Connector for CaliberRM	5.0.0.685
☐ pure::variants - Connector for Ecore	5.0.0.685

Details

The pure::variants plugin enables variant management for software product lines.

☑ Show only the latest versions of available software

☐ Hide items that are already installed

☑ Group items by category

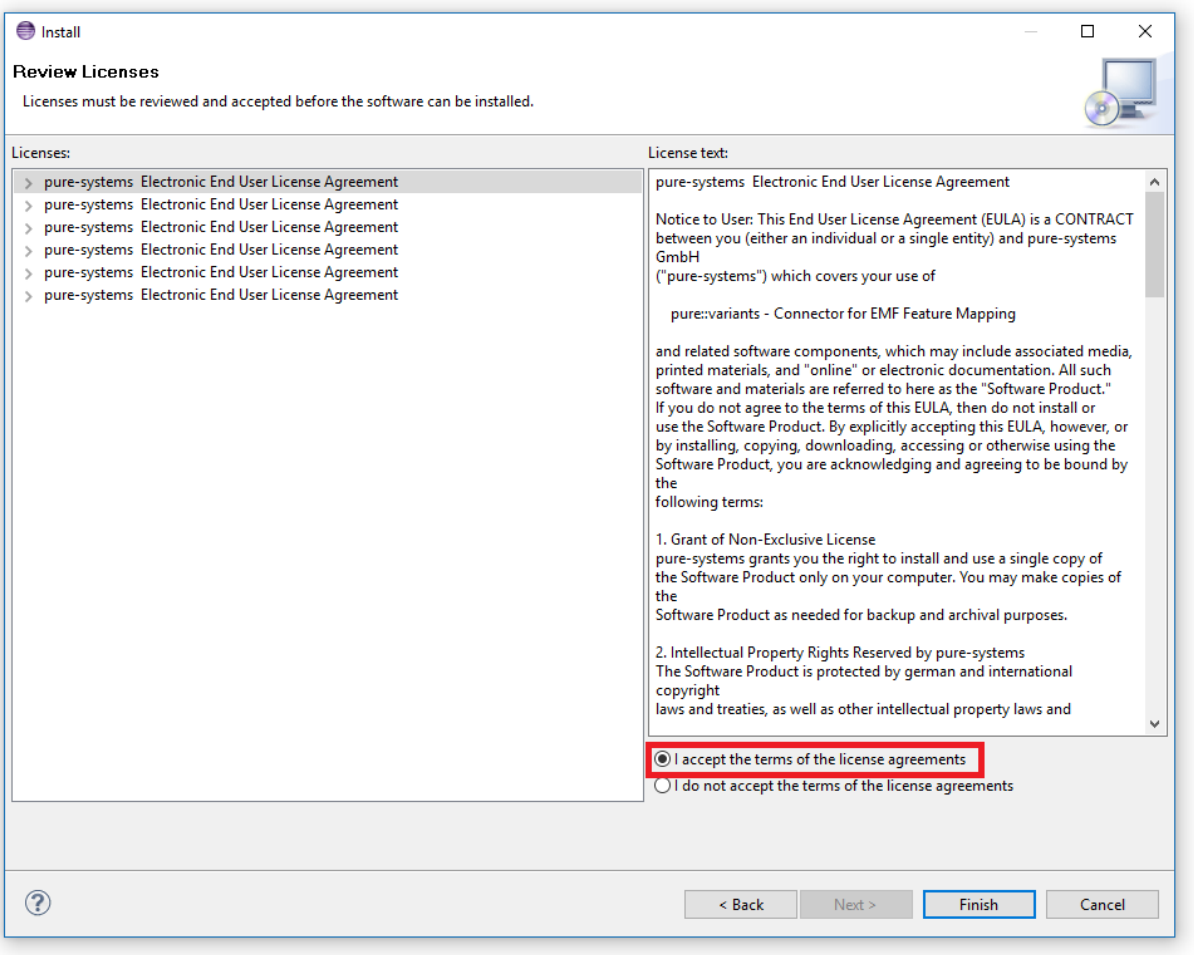
What is already installed?

☐ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

- Accept license, hit "Next" and then "Finish".

Figure 10. Licence Agreement



Review Licenses

Licenses must be reviewed and accepted before the software can be installed.

Licenses:

- pure-systems Electronic End User License Agreement
- pure-systems Electronic End User License Agreement
- pure-systems Electronic End User License Agreement
- pure-systems Electronic End User License Agreement
- pure-systems Electronic End User License Agreement
- pure-systems Electronic End User License Agreement

License text:

pure-systems Electronic End User License Agreement

Notice to User: This End User License Agreement (EULA) is a CONTRACT between you (either an individual or a single entity) and pure-systems GmbH ("pure-systems") which covers your use of

pure::variants - Connector for EMF Feature Mapping

and related software components, which may include associated media, printed materials, and "online" or electronic documentation. All such software and materials are referred to here as the "Software Product." If you do not agree to the terms of this EULA, then do not install or use the Software Product. By explicitly accepting this EULA, however, or by installing, copying, downloading, accessing or otherwise using the Software Product, you are acknowledging and agreeing to be bound by the following terms:

1. Grant of Non-Exclusive License
pure-systems grants you the right to install and use a single copy of the Software Product only on your computer. You may make copies of the Software Product as needed for backup and archival purposes.

2. Intellectual Property Rights Reserved by pure-systems
The Software Product is protected by german and international copyright laws and treaties, as well as other intellectual property laws and

[✓ I accept the terms of the license agreements]
I do not accept the terms of the license agreements

- In the dialog select "Install all".
- Restart pure::variants when asked for.

If the direct remote update is not possible (often due to firewall/proxies preventing Eclipse accessing external web sites), please go to the web site using an Internet browser:

- For pure::variants Evaluation use https://www.pure-systems.com/pv-update
- For pure::variants Enterprise use https://www.pure-systems.com/pvde-update

and download the "Complete Updatesite" archive:

- Start pure::variants (or the Eclipse into which pure::variants has been installed).
- Select "Help"->"Software Updates"->"Find and Install...".
- Select "Search for new features to install" and "Next".
- Click on button "Archived Update Site" or "Local Update Site".
- Use "Browse" to select the downloaded archive file.
- Press "Ok". The pure::variants update site from the archive should be selected.
- All other check boxes should be unselected to speed up the process. Press "Finish".

- Unfold everything below pure::variants update site and select all features to be updated. Select "Next".
- Accept license, hit "Next" and then "Finish".
- In the dialog, select "Install all".
- Restart pure::variants when asked for.

3.2. Update pure::variants Client

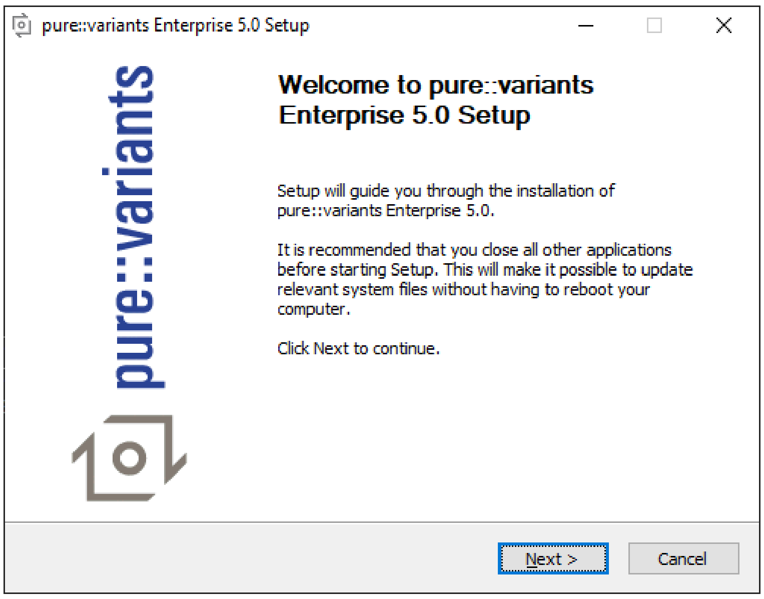
3.2.1. Update with pure::variants Installer

This update method is available for Windows only. If you do not use Windows please see Section 3.2.2, “Update with Update Action” or Section 3.2.3, “Update with Eclipse package manager”.

The Windows Installer can be downloaded from the pure::variants product web page. Go to http://www.pure-systems.com/pvde-update. The product download pages are protected by a password. You need to login using the email address and the registration number from the license file.

Download the installer package ("pure::variants Windows Installer Package") and extract it. The installer will check for an existing pure::variants client installation and start in update mode if it finds one. Start the update by double-clicking "Setup Enterprise X.Y.ZZ.exe". Running the pure::variants enterprise installer requires administrator privileges.

Figure 11. pure::variants Client Installer



Welcome to pure::variants Enterprise 5.0 Setup

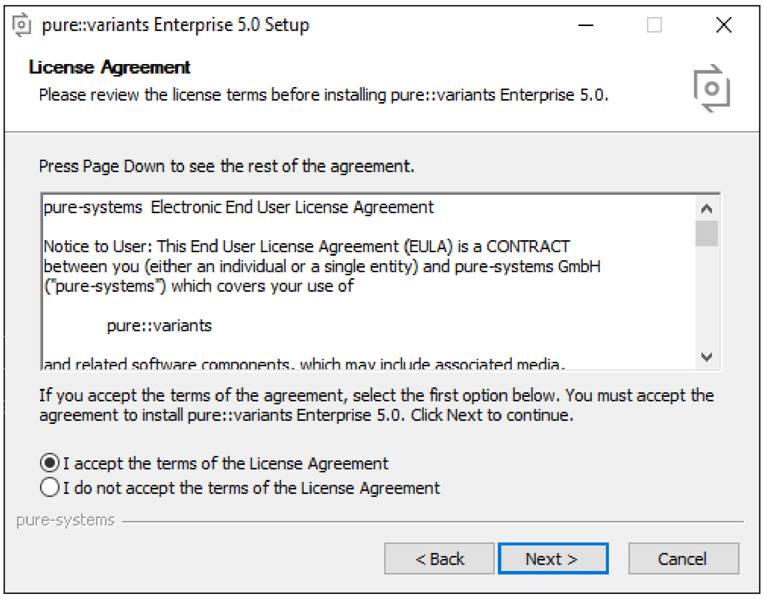
Setup will guide you through the installation of pure::variants Enterprise 5.0.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Click *Next*.

Figure 12. Setup pure::variants Client License


pure::variants Enterprise 5.0 Setup

License Agreement

Please review the license terms before installing pure::variants Enterprise 5.0.

Press Page Down to see the rest of the agreement.

pure-systems Electronic End User License Agreement

Notice to User: This End User License Agreement (EULA) is a CONTRACT between you (either an individual or a single entity) and pure-systems GmbH ("pure-systems") which covers your use of

pure::variants

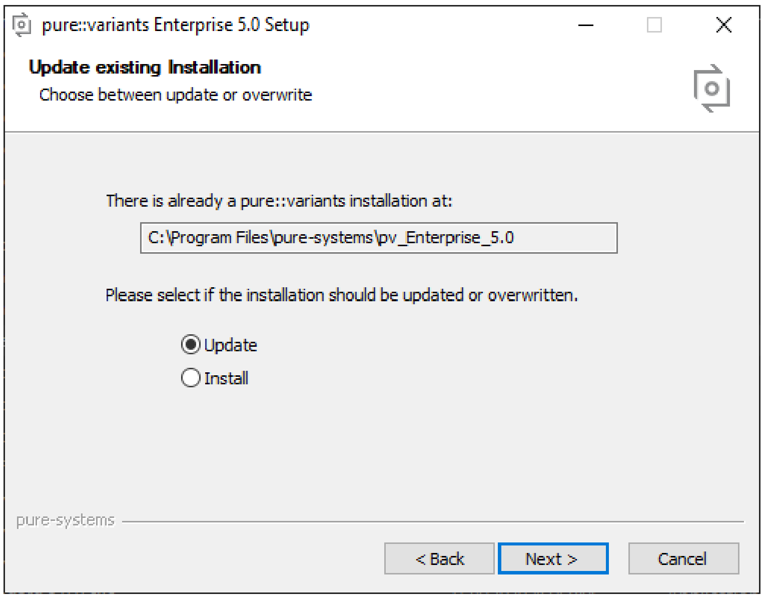
and related software components, which may include associated media,

If you accept the terms of the agreement, select the first option below. You must accept the agreement to install pure::variants Enterprise 5.0. Click Next to continue.

[✓ I accept the terms of the License Agreement]
I do not accept the terms of the License Agreement

Read the license agreement, and after accepting it click *Next*.

Figure 13. Choose Update Mode


pure::variants Enterprise 5.0 Setup

Update existing Installation

Choose between update or overwrite

There is already a pure::variants installation at:

C:\Program Files\pure-systems\pv_Enterprise_5.0

Please select if the installation should be updated or overwritten.

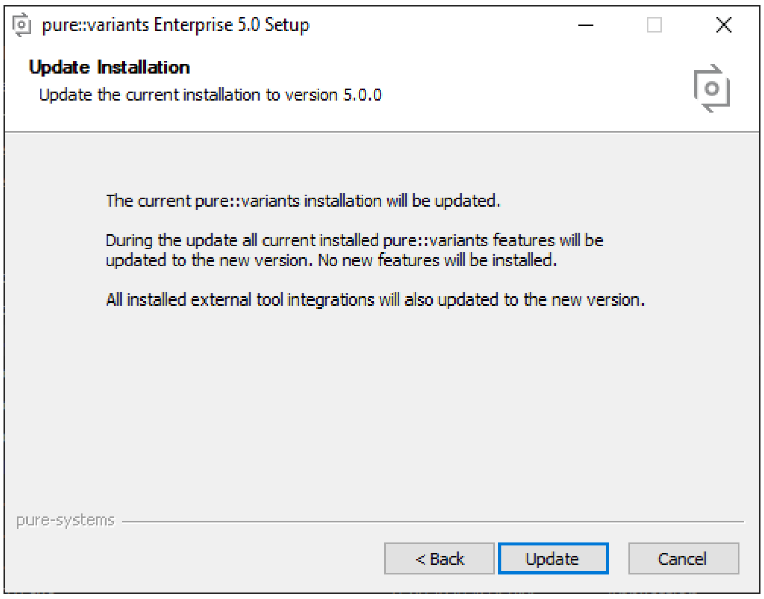
[✓ Update]
Install

Choose *Update* if the current pure::variants client installation shall just be updated with the same installed feature and settings. The installed pure::variants integrations will also be updated. The installed components cannot be changed. If a change of the installed components is wanted, choose *Install* mode.

Or choose *Install* if the current pure::variants installation shall be removed and a new fresh pure::variants client shall be installed. The *Install* option runs the installer as described in Section 3.1.1, “Install with pure::variants Installer”. Please see this section for further installation steps.

Click *Next*.

Figure 14. pure::variants Start Update



Update Installation
Update the current installation to version 5.0.0

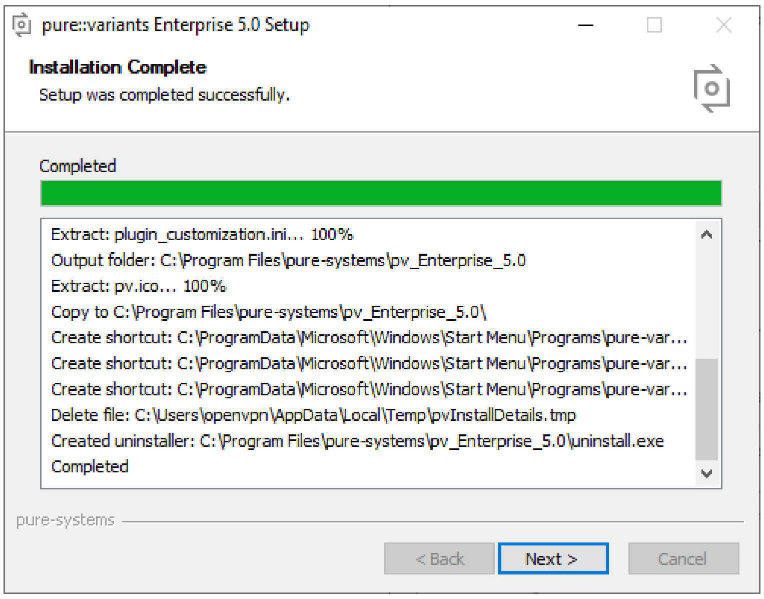
The current pure::variants installation will be updated.

During the update all current installed pure::variants features will be updated to the new version. No new features will be installed.

All installed external tool integrations will also updated to the new version.

Click *Update* to start the update process.

Figure 15. pure::variants Installation Progress



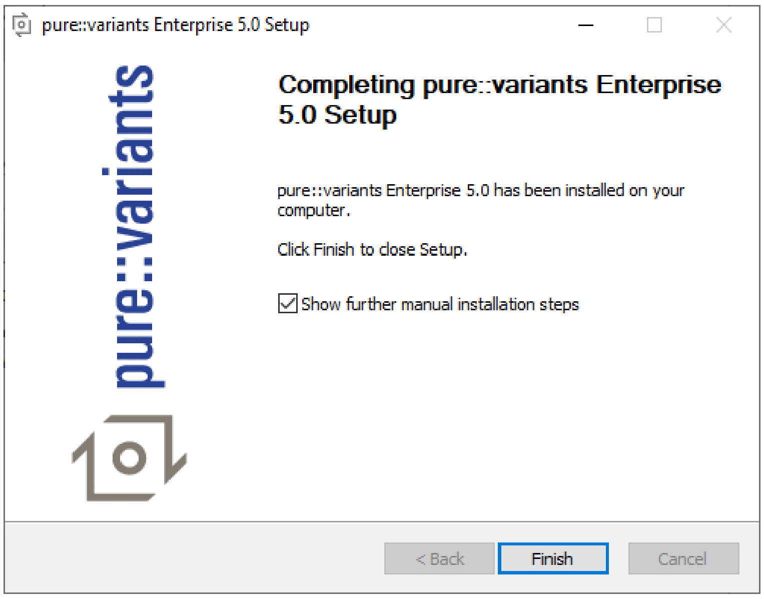
Installation Complete
Setup was completed successfully.

Completed

```
Extract: plugin_customization.ini... 100%
Output folder: C:\Program Files\pure-systems\pv_Enterprise_5.0
Extract: pv.ico... 100%
Copy to C:\Program Files\pure-systems\pv_Enterprise_5.0\
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\pure-var...
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\pure-var...
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\pure-var...
Delete file: C:\Users\openvpn\AppData\Local\Temp\pvInstallDetails.tmp
Created uninstaller: C:\Program Files\pure-systems\pv_Enterprise_5.0\uninstall.exe
Completed
```

This page is showing the installation details. Click *Next* after this is finished.

Figure 16. Update pure::variants Client Finish Page



Completing pure::variants Enterprise 5.0 Setup

pure::variants Enterprise 5.0 has been installed on your computer.

Click Finish to close Setup.

☑ Show further manual installation steps

The Option *Show further manual installation steps* will open a text document showing more information about the installed integrations and possible manual installation steps, which have to be performed for the integrations to work properly.

3.2.2. Update with Update Action

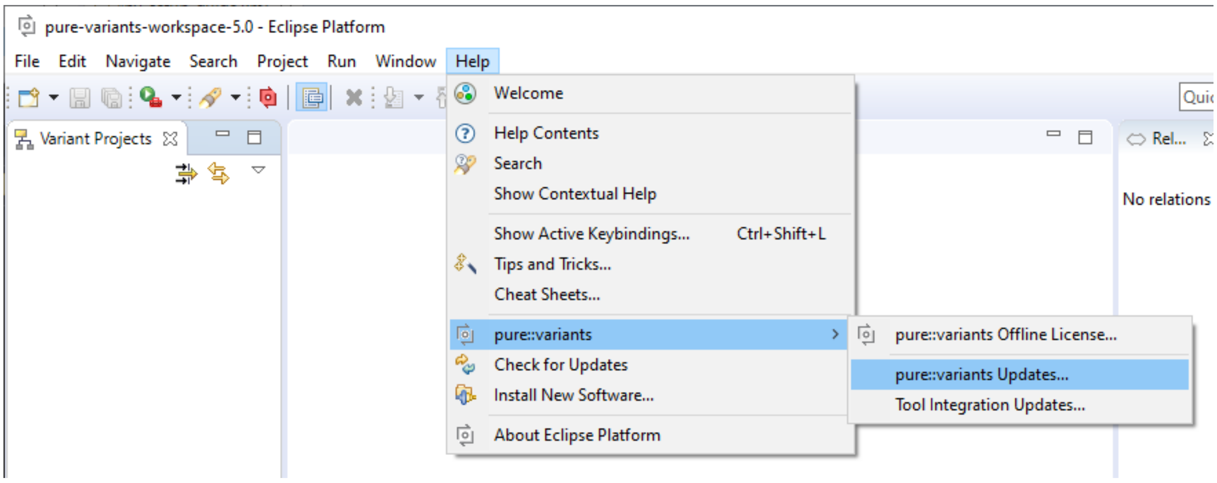
pure::variants has a built-in update action which can be used to perform an update with all the currently installed pure::variants extensions. This update action does not update the installed pure::variants integrations automatically. But they can be easily updated with the *Tool Integration Update* action. See Section 7.2, “Update pure::variants Tool Integrations” for the detailed description.

The update action requires administrator privileges.

Note

The pure::variants client restarts automatically after the update process finished. So please make sure that all open editors are saved and closed before continuing.

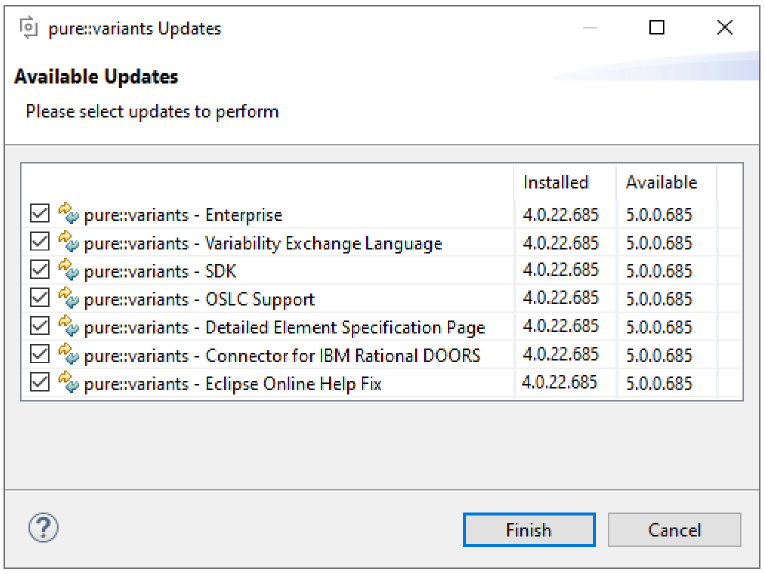
Figure 17. Start pure::variants Client Update


pure-variants-workspace-5.0 - Eclipse Platform
File Edit Navigate Search Project Run Window Help
Variant Projects
Rel...
No relations
Welcome
Help Contents
Search
Show Contextual Help
Show Active Keybindings... Ctrl+Shift+L
Tips and Tricks...
Cheat Sheets...
pure::variants
Check for Updates
Install New Software...
About Eclipse Platform
pure::variants Offline License...
pure::variants Updates...
Tool Integration Updates...

Start the pure::variants client update with the *pure::variants Updates...* action from the pure::variants *Help* menu. The action can be found in the *pure::variants* sub-menu.

If the pure::variants client is not started as Administrator, a dialog comes up to inform that pure::variants has to be started as Administrator.

Figure 18. Start pure::variants Client Update


pure::variants Updates

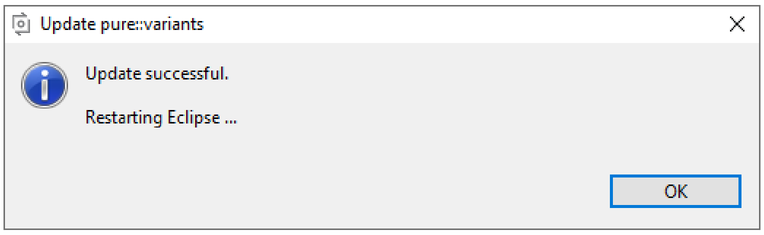
Available Updates

Please select updates to perform

	Installed	Available
☑ pure::variants - Enterprise	4.0.22.685	5.0.0.685
☑ pure::variants - Variability Exchange Language	4.0.22.685	5.0.0.685
☑ pure::variants - SDK	4.0.22.685	5.0.0.685
☑ pure::variants - OSLC Support	4.0.22.685	5.0.0.685
☑ pure::variants - Detailed Element Specification Page	4.0.22.685	5.0.0.685
☑ pure::variants - Connector for IBM Rational DOORS	4.0.22.685	5.0.0.685
☑ pure::variants - Eclipse Online Help Fix	4.0.22.685	5.0.0.685

A dialog comes up and shows all available updates. Select the features to update an click *Finish*. The update process starts and shows the progress in the same window.

Figure 19. Start pure::variants Client Update


Update pure::variants

Update successful.

Restarting Eclipse ...

After the update process finished, the pure::variants client restarts automatically.

3.2.3. Update with Eclipse package manager

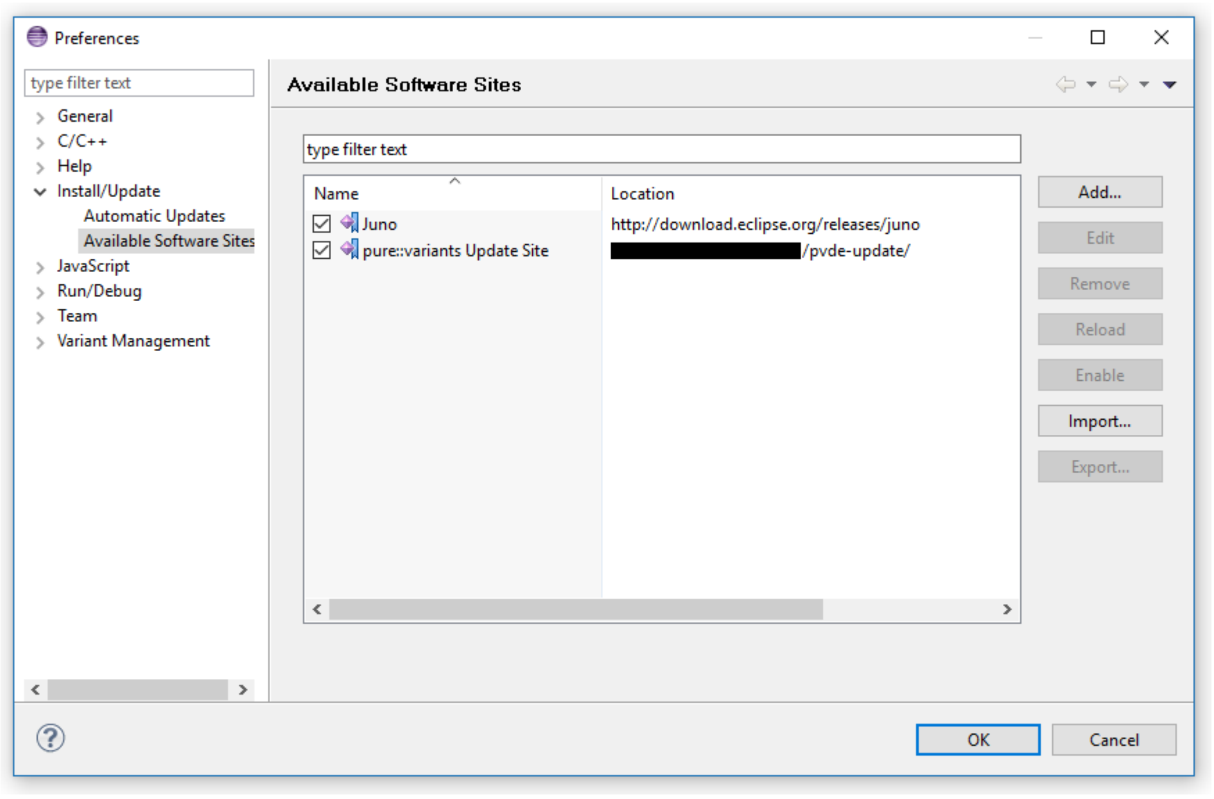
The quickest way to get a update for pure:.variants is to run the software updater inside pure::variants:

- Start pure::variants (or the Eclipse into which pure::variants has been installed).
- Select "Help"->"Install New Software...".
- Select "pure::variants update site" from the available Software Sites.

  If location "pure::variants update site" is not present, enter your location in the edit field, or press "Add" if you have a local update site at hand.

  The location of the site depends on the pure::variants product variant. Visit the pure-systems web site (http://www.pure-systems.com/pv) or read your registration email to find out which site is relevant for the version of the software you are using.

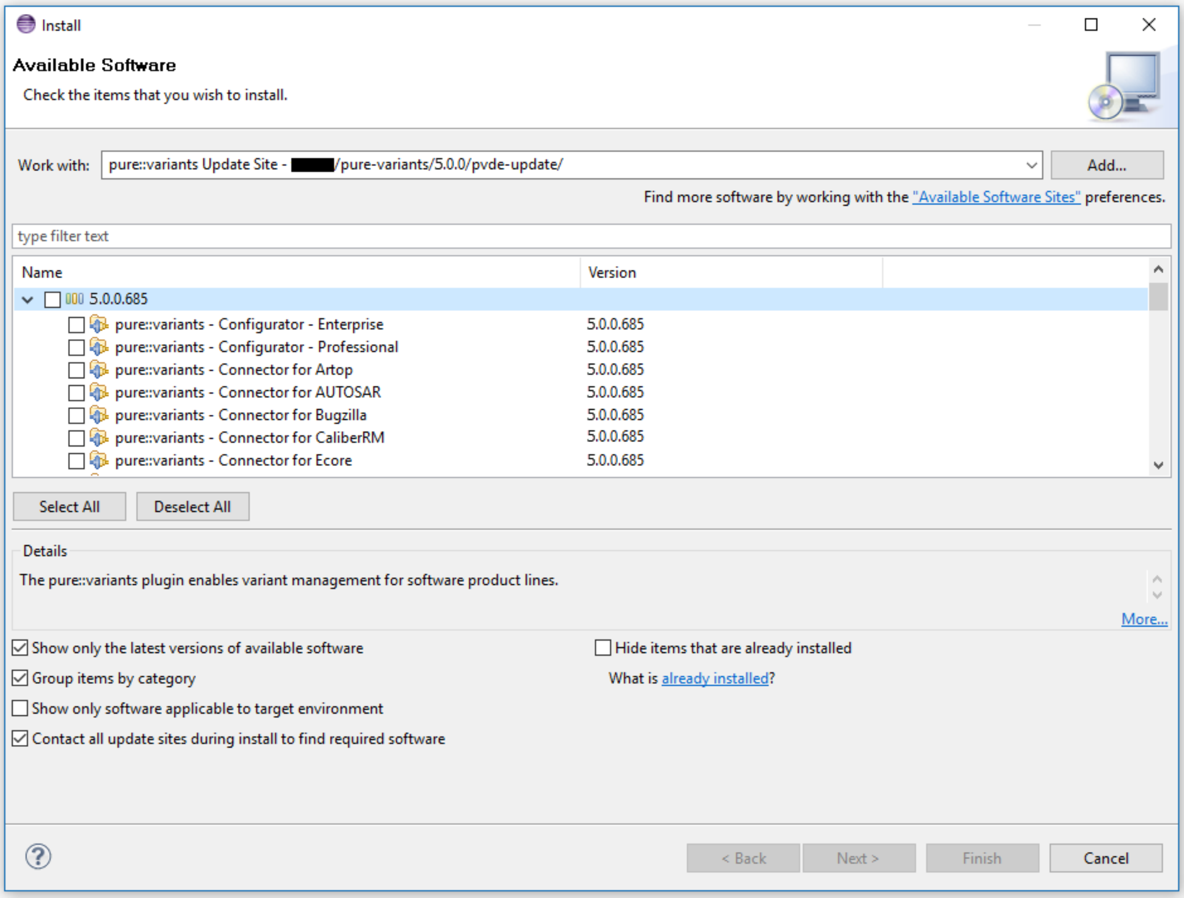
Figure 20. Update Site Selection



Name	Location
☑ Juno	http://download.eclipse.org/releases/juno
☑ pure::variants Update Site	[redacted]/pvde-update/

- Unfold the pure::variants update site and select all features to be updated. Select "Next".

Figure 21. pure::variants Plugin Selection



Install

Available Software

Check the items that you wish to install.

Work with: pure::variants Update Site - [redacted]/pure-variants/5.0.0/pvde-update/

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☐ 5.0.0.685	
☐ pure::variants - Configurator - Enterprise	5.0.0.685
☐ pure::variants - Configurator - Professional	5.0.0.685
☐ pure::variants - Connector for Artop	5.0.0.685
☐ pure::variants - Connector for AUTOSAR	5.0.0.685
☐ pure::variants - Connector for Bugzilla	5.0.0.685
☐ pure::variants - Connector for CaliberRM	5.0.0.685
☐ pure::variants - Connector for Ecore	5.0.0.685

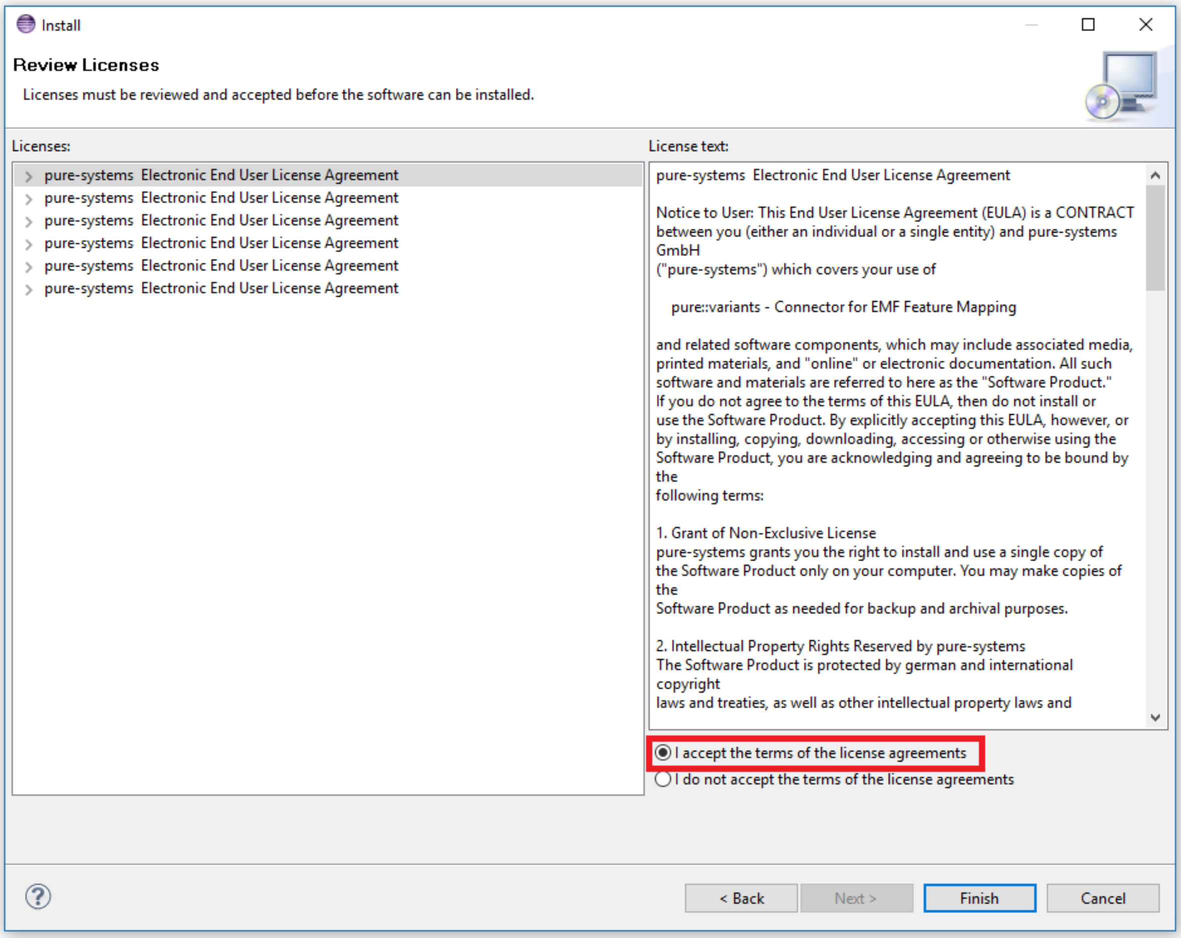
Details

The pure::variants plugin enables variant management for software product lines.

☑ Show only the latest versions of available software
☑ Group items by category
☐ Show only software applicable to target environment
☑ Contact all update sites during install to find required software
☐ Hide items that are already installed
What is already installed?

- Accept the license, hit "Next" and then "Finish".

Figure 22. Licence Agreement


Install

Review Licenses

Licenses must be reviewed and accepted before the software can be installed.

Licenses:

- pure-systems Electronic End User License Agreement
- pure-systems Electronic End User License Agreement
- pure-systems Electronic End User License Agreement
- pure-systems Electronic End User License Agreement
- pure-systems Electronic End User License Agreement
- pure-systems Electronic End User License Agreement

License text:

pure-systems Electronic End User License Agreement

Notice to User: This End User License Agreement (EULA) is a CONTRACT between you (either an individual or a single entity) and pure-systems GmbH ("pure-systems") which covers your use of

pure::variants - Connector for EMF Feature Mapping

and related software components, which may include associated media, printed materials, and "online" or electronic documentation. All such software and materials are referred to here as the "Software Product." If you do not agree to the terms of this EULA, then do not install or use the Software Product. By explicitly accepting this EULA, however, or by installing, copying, downloading, accessing or otherwise using the Software Product, you are acknowledging and agreeing to be bound by the following terms:

1. Grant of Non-Exclusive License
pure-systems grants you the right to install and use a single copy of the Software Product only on your computer. You may make copies of the Software Product as needed for backup and archival purposes.

2. Intellectual Property Rights Reserved by pure-systems
The Software Product is protected by german and international copyright laws and treaties, as well as other intellectual property laws and

[✓ I accept the terms of the license agreements]
I do not accept the terms of the license agreements

- In the dialog select "Install all".
- Restart pure::variants when asked for.

If the online update is not possible (often due to firewall/proxies preventing Eclipse accessing external web sites), please go to the web site using an Internet browser:

- For pure::variants Evaluation use https://www.pure-systems.com/pv-update
- For pure::variants Enterprise/Professional use https://www.pure-systems.com/pvde-update

and download the "Complete Updatesite" archive:

- Start pure::variants (or the Eclipse into which pure::variants has been installed).
- Select "Help"->"Software Updates"->"Find and Install...".
- Select "Search for new features to install" and "Next".
- Click on button "Archived Update Site" or "Local Update Site".
- Use "Browse" to select the downloaded archive file.
- Press "Ok". The pure::variants update site from the archive should be selected.
- All other check boxes should be unselected to speed up the process. Press "Finish".

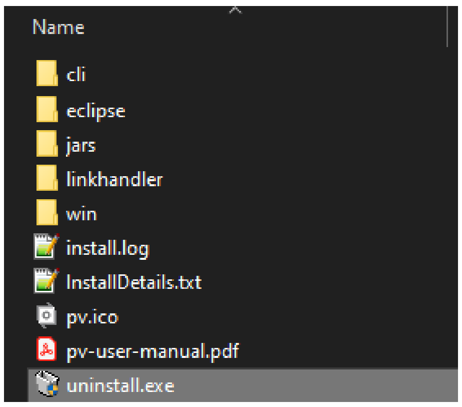
- Unfold everything below pure::variants update site and select the features to be updated. Select "Next".
- Accept the license, hit "Next" and then "Finish".
- In the dialog select "Install all".
- Restart pure::variants when asked for.

3.3. Uninstall pure::variants Client

3.3.1. Uninstall using pure::variants Uninstaller

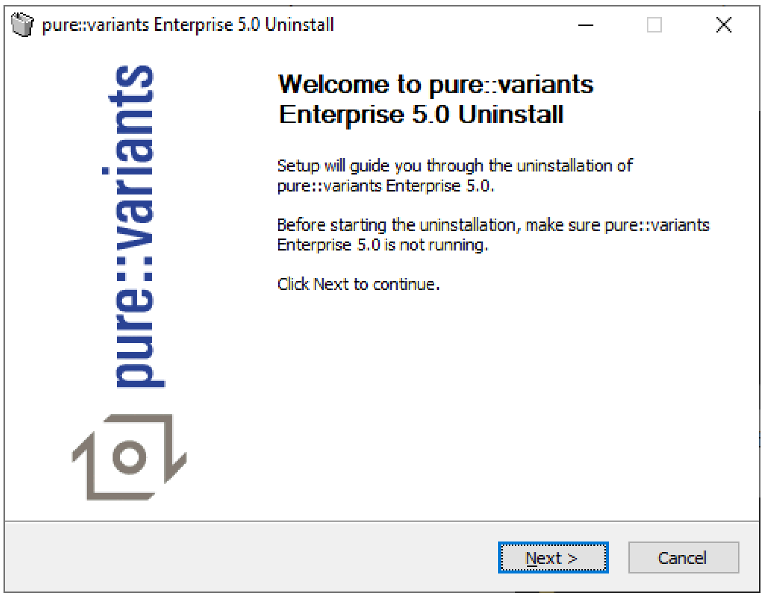
The uninstaller for the pure::variants client can be started in two different ways. The first is to go to the Windows *Add or remove programs* application and search for *pure::variants Enterprise* and start the uninstaller by using the *Uninstall* action. The uninstaller requires Administrator privileges.

Figure 23. pure::variants Client Uninstaller



The second way is to navigate to the pure::variants client installation folder and start the uninstaller by double clicking it.

Figure 24. pure::variants Client Uninstaller



Welcome to pure::variants Enterprise 5.0 Uninstall

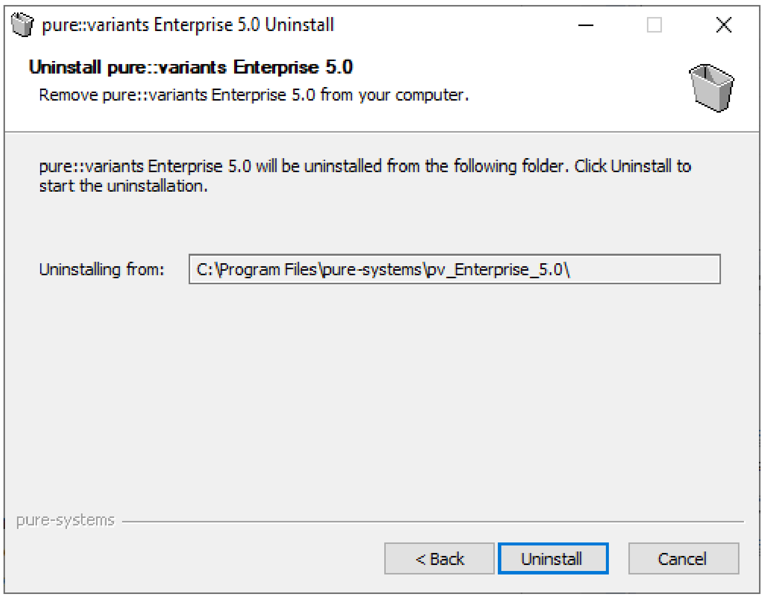
Setup will guide you through the uninstallation of pure::variants Enterprise 5.0.

Before starting the uninstallation, make sure pure::variants Enterprise 5.0 is not running.

Click Next to continue.

Click *Next*.

Figure 25. Uninstall from


pure::variants Enterprise 5.0 Uninstall

Uninstall pure::variants Enterprise 5.0

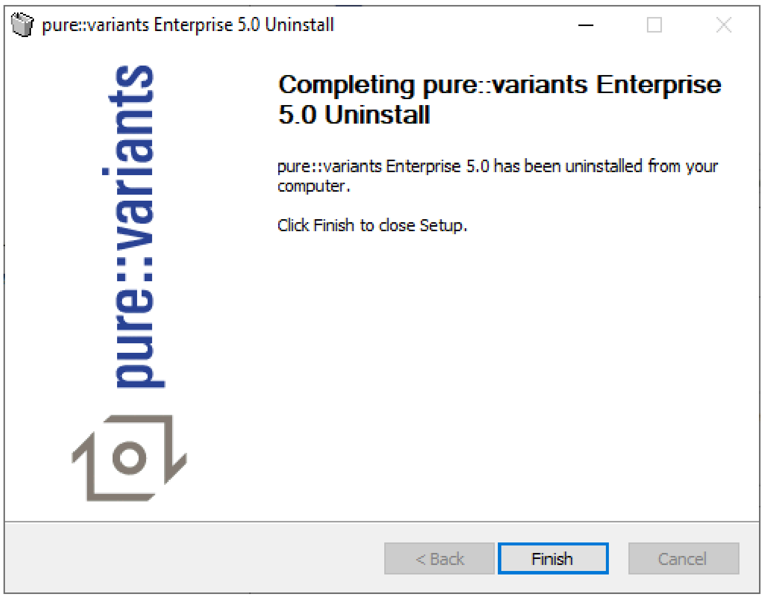
Remove pure::variants Enterprise 5.0 from your computer.

pure::variants Enterprise 5.0 will be uninstalled from the following folder. Click Uninstall to start the uninstallation.

Uninstalling from: C:\Program Files\pure-systems\pv_Enterprise_5.0\

Click *Uninstall* to start the uninstall process.

Figure 26. Completing Uninstall


pure::variants Enterprise 5.0 Uninstall

Completing pure::variants Enterprise 5.0 Uninstall

pure::variants Enterprise 5.0 has been uninstalled from your computer.

Click Finish to close Setup.

Click *Finish* to close the uninstaller.

3.3.2. Uninstall pure::variants from existing Eclipse instance

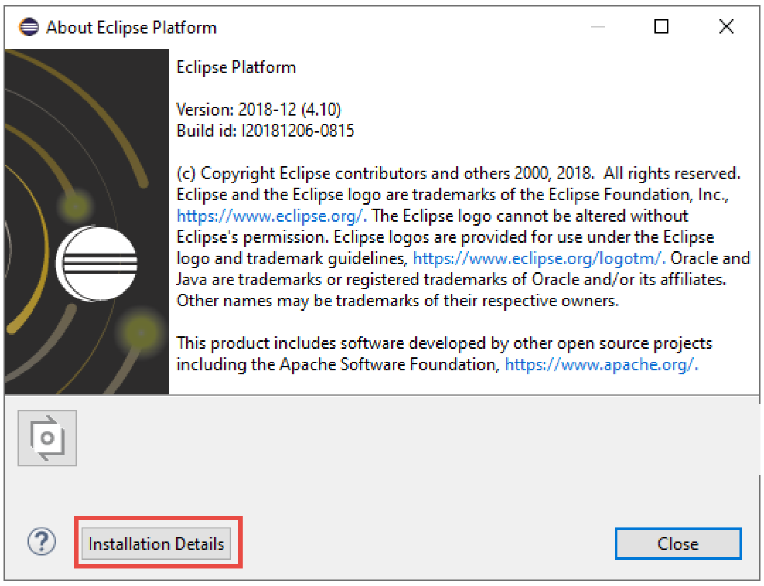
There are two ways to remove pure::variants from an Eclipse instance. You can use the Eclipse command line or remove the pure::variants features one by one in the running Eclipse Instance. Either way a cleanup of the Eclipse instance has to be performed afterwards.

If the Eclipse instance is not needed anymore you can just remove the whole Eclipse installation from the file system. If the Eclipse is of further use, follow one of the installation methods.

Uninstall pure::variants in running Eclipse Instance

To remove an installed feature from Eclipse using the Eclipse client, open the *About Eclipse Platform* dialog with the *About Eclipse Platform* action in the *Help* menu.

Figure 27. Eclipse About Dialog



Eclipse Platform

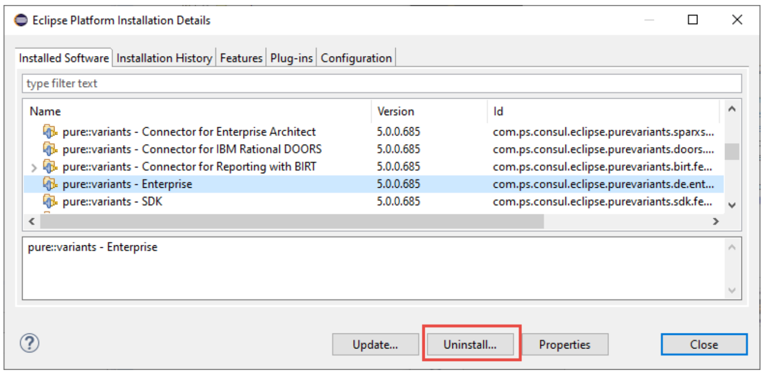
Version: 2018-12 (4.10)
Build id: I20181206-0815

(c) Copyright Eclipse contributors and others 2000, 2018. All rights reserved. Eclipse and the Eclipse logo are trademarks of the Eclipse Foundation, Inc., https://www.eclipse.org/. The Eclipse logo cannot be altered without Eclipse's permission. Eclipse logos are provided for use under the Eclipse logo and trademark guidelines, https://www.eclipse.org/logotm/. Oracle and Java are trademarks or registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This product includes software developed by other open source projects including the Apache Software Foundation, https://www.apache.org/.

Use the *Installation Details* button to access the installation details.

Figure 28. Eclipse About Dialog

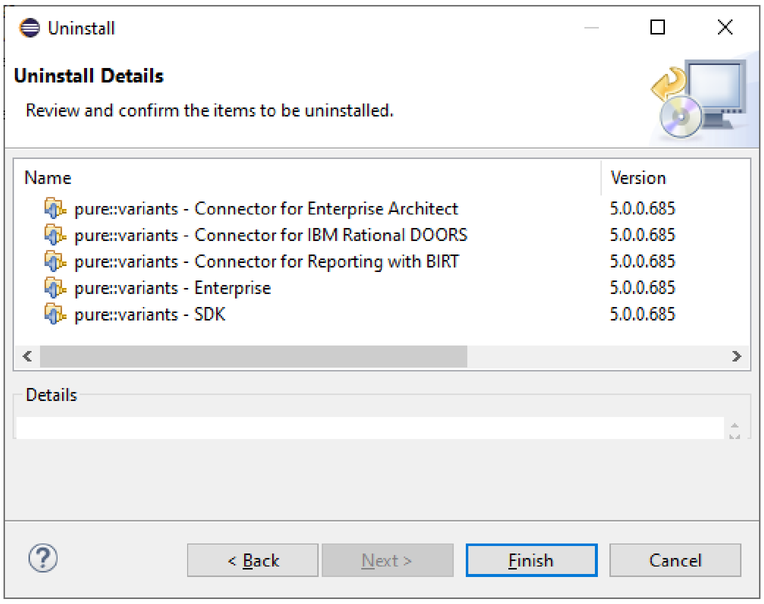


Name	Version	Id
pure::variants - Connector for Enterprise Architect	5.0.0.685	com.ps.consul.eclipse.purevariants.sparxs...
pure::variants - Connector for IBM Rational DOORS	5.0.0.685	com.ps.consul.eclipse.purevariants.doors....
pure::variants - Connector for Reporting with BIRT	5.0.0.685	com.ps.consul.eclipse.purevariants.birt.fe...
pure::variants - Enterprise	5.0.0.685	com.ps.consul.eclipse.purevariants.de.ent...
pure::variants - SDK	5.0.0.685	com.ps.consul.eclipse.purevariants.sdk.fe...

pure::variants - Enterprise

Use the *Uninstall* button to start the uninstallation of the selected features. Selecting multiple features at once is possible.

Figure 29. Eclipse About Dialog



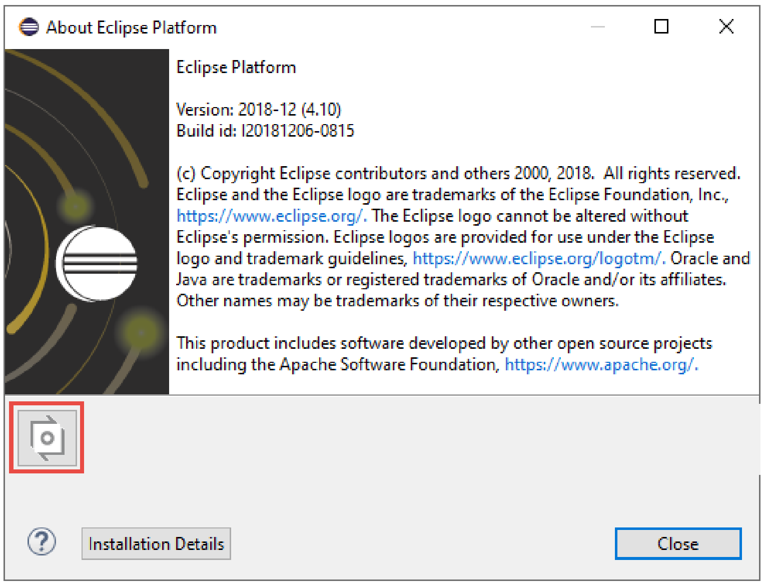
Name	Version
pure::variants - Connector for Enterprise Architect	5.0.0.685
pure::variants - Connector for IBM Rational DOORS	5.0.0.685
pure::variants - Connector for Reporting with BIRT	5.0.0.685
pure::variants - Enterprise	5.0.0.685
pure::variants - SDK	5.0.0.685

Click *Finish* to start the uninstall process. After it finished, Eclipse will prompt you to restart the application. Click *Restart* to finish the uninstallation.

Uninstall pure::variants using Eclipse uninstall application

To use the uninstall application you need the feature ids of the features you want to uninstall. The feature ids can be found in the *About Eclipse Plattform* dialog. Open the dialog with the *About Eclipse Platform* action in the *Help* menu.

Figure 30. Eclipse About Dialog



Eclipse Platform

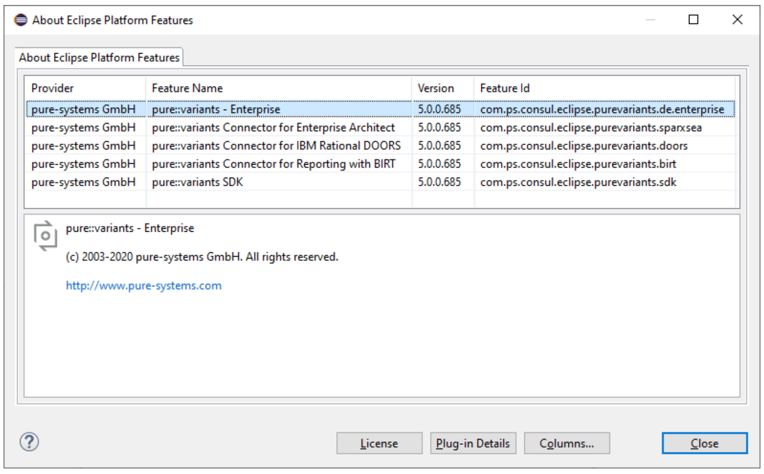
Version: 2018-12 (4.10)
Build id: I20181206-0815

(c) Copyright Eclipse contributors and others 2000, 2018. All rights reserved. Eclipse and the Eclipse logo are trademarks of the Eclipse Foundation, Inc., https://www.eclipse.org/. The Eclipse logo cannot be altered without Eclipse's permission. Eclipse logos are provided for use under the Eclipse logo and trademark guidelines, https://www.eclipse.org/logotm/. Oracle and Java are trademarks or registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This product includes software developed by other open source projects including the Apache Software Foundation, https://www.apache.org/.

Click on the pure::variants icon.

Figure 31. Installed pure::variants features



About Eclipse Platform Features

Provider	Feature Name	Version	Feature Id
pure-systems GmbH	pure::variants - Enterprise	5.0.0.685	com.ps.consul.eclipse.purevariants.de.enterprise
pure-systems GmbH	pure::variants Connector for Enterprise Architect	5.0.0.685	com.ps.consul.eclipse.purevariants.sparxsea
pure-systems GmbH	pure::variants Connector for IBM Rational DOORS	5.0.0.685	com.ps.consul.eclipse.purevariants.doors
pure-systems GmbH	pure::variants Connector for Reporting with BIRT	5.0.0.685	com.ps.consul.eclipse.purevariants.birt
pure-systems GmbH	pure::variants SDK	5.0.0.685	com.ps.consul.eclipse.purevariants.sdk

pure::variants - Enterprise

(c) 2003-2020 pure-systems GmbH. All rights reserved.

http://www.pure-systems.com

The feature ids are listed in the *Feature Id* column of the upcoming dialog. All feature ids have to be extended by ".feature.group" and are concatenated with ",". The feature id list for the example shown in the previous figure would be:

```
com.ps.consul.eclipse.purevariants.sparxsea.feature.group,com.ps.consul.eclipse.purevariants.
birt.feature.group,com.ps.consul.eclipse.purevariants.de.enterprise.feature.group,com.ps.
consul.eclipse.purevariants.doors.feature.group,com.ps.consul.eclipse.purevariants.sdk.
feature.group
```

The resulting list of feature ids is used in the following command.

```
"<Eclipse Installation Directory>\eclipsec.exe" -nosplash --launcher.suppressErrors -
application
org.eclipse.equinox.p2.director -uninstallIU "<list of feature ids>" -data "ws" -vmargs
-Dequinox.ds.block_timeout=120000
-Dorg.eclipse.ecf.provider.filetransfer.retrieve.readTimeout=120000
-Declipse.p2.mirrors=false -Xms100m -Xmx2048m -Xmnx64m -Xgcpolicy:gencon
-XX:MaxPermSize=512M -Xcompressedrefs
```

Cleanup Eclipse after uninstallation

pure::variants stores some settings, license and log files at two locations in the file system. On Windows the first one is C:\Users\<user name>\AppData\Roaming\pure-variants-5, and the second C:\ProgramData\pure-variants-5. On Linux based systems the pure-variants-5 folders are located in the users home directory and at */usr/share*. These folders should be removed after pure::variants has been completely removed from the computer.

To clean up the Eclipse instance, run the following command.

```
"<Eclipse Installation Directory>\eclipsec.exe" -nosplash --launcher.suppressErrors -
application
org.eclipse.equinox.p2.garbagecollector.application -data "ws" -vmargs
-Dequinox.ds.block_timeout=120000
-Dorg.eclipse.ecf.provider.filetransfer.retrieve.readTimeout=120000
-Declipse.p2.mirrors=false -Xms100m -Xmx2048m -Xmnx64m -Xgcpolicy:gencon
-XX:MaxPermSize=512M -Xcompressedrefs
```

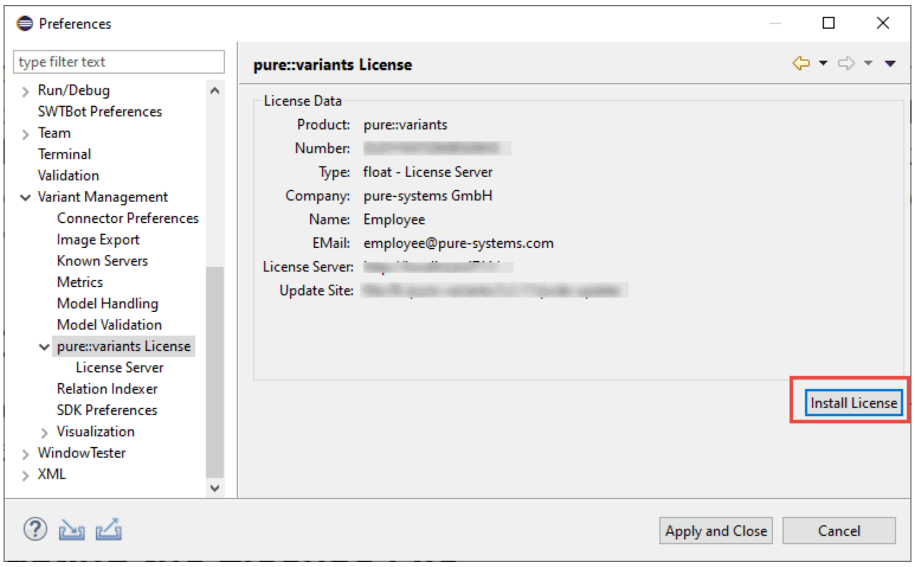
3.4. Basic Setup of the pure::variants Client

3.4.1. Setup a pure::variants Client License

A valid license file is required in order to use pure::variants. If pure::variants is started and no license is present, then the user is prompted to supply a license. Select the “Yes” button and use the file dialog to specify the license file delivered with pure::variants. The specified license will be stored in the user's application data directory. If you are using multiple workspaces then the license file has to be installed only once. The pure::variants integrations also use the installed license and thus no further setup step is needed here.

To replace an existing pure::variants license, start pure::variants and open the **Preferences** (menu Window -> Preferences). Navigate to **Variant Management -> pure::variants License** and use the **Install License** button to select the new license.

Figure 32. pure::variants License Preferences

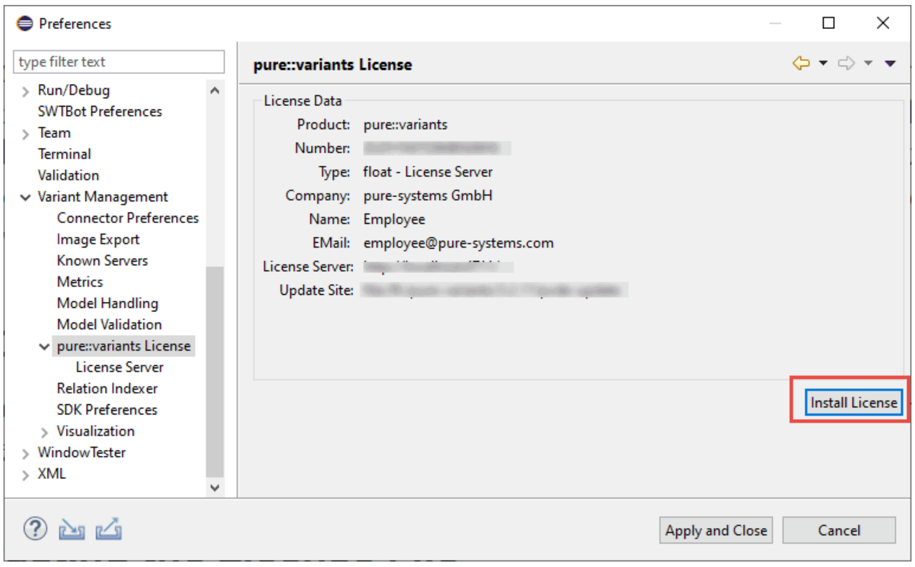


Product: pure::variants
Number: [redacted]
Type: float - License Server
Company: pure-systems GmbH
Name: Employee
EMail: employee@pure-systems.com
License Server: [redacted]
Update Site: [redacted]

3.4.2. Update a pure::variants Client License

If pure::variants is not explicitly asking for a new license, the update can be forced by starting pure::variants and opening menu **Window -> Preferences**. Select **Variant Management -> pure::variants License** and install the license using the provided **Install** button.

Figure 33. pure::variants License Preferences



Product: pure::variants
Number: [redacted]
Type: float - License Server
Company: pure-systems GmbH
Name: Employee
EMail: employee@pure-systems.com
License Server: [redacted]
Update Site: [redacted]

3.4.3. Add pure::variants Client License using environment variable or Java property

For central or automatic deployed pure::variants clients it may be necessary to also automatically deploy or update the pure::variants client license. For this use case the variable **PVLICENSE** can be used. This variable can either be introduced as an environment variable or just added as a Java property to the command line starting pure::variants. If this variable is set, the given license is used instead of a possibly previously installed license.

Example for the command line parameter: *-DPVLICENSE=C:/absolute/path/to/the/license/file.lic*

3.5. Trouble Shooting

3.5.1. pure::variants is low on memory

If pure::variants is low on memory it can result in out of memory errors or causing pure::variants to run very slow since Java is trying to free up memory constantly by running the garbage collector.

To solve that problem pure::variants needs to be enabled to use more memory. This can be done by editing the eclipse.ini file, which is located in <pure::variants installation path>\eclipse\eclipse.ini.

Add the following three lines to the end of the ini file, if not existing yet. The first line tells Eclipse that there are Java Virtual Machine options following. Xms defines the minimal amount of memory Java is reserving. Xmx defines the maximum amount of memory Java is allowed to use. The default value is 1024 MB. We recommend to set the value to 6144 MB .

```
-vmargs
-Xms40m
-Xmx6144m
```

Note

If Eclipse does not start after the eclipse.ini was changed, the maximum amount of memory defined is not valid. There are multiple reasons for this, e.g. Java could not reserve enough memory. Try to decrease the defined maximum memory.

4. pure::variants License Server

There are two alternative ways to install pure::variants License Server, depending on your operating system. The different installation procedures are described below.

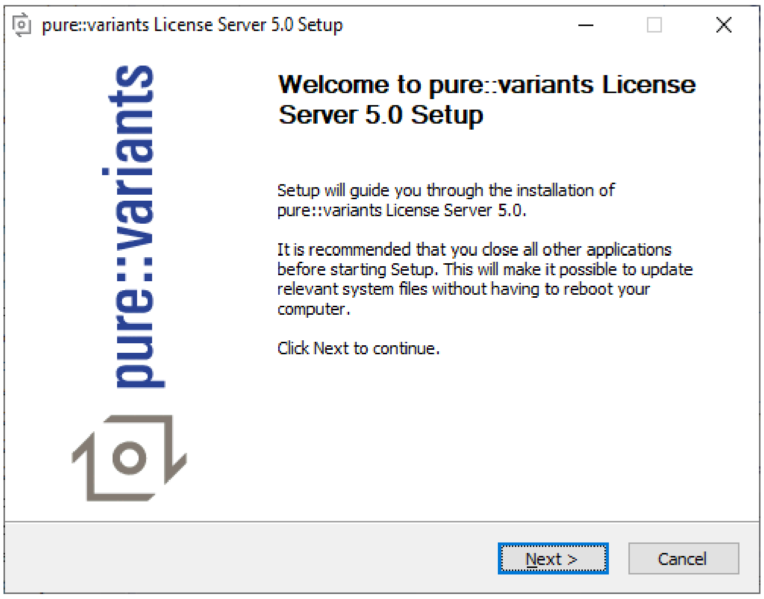
4.1. Install pure::variants License Server

4.1.1. Install with Windows Installer (Windows only)

The Windows Installer can be downloaded from the pure::variants product web page. Go to http://www.pure-systems.com/pvde-update. The product download pages are protected by a password. You need to login by using the email address and the registration number from the license file.

Download the license server installer “Setup License Server X.Y.ZZ.exe” and start the installation by double-clicking the executable. Running the license server installer requires Administrator privileges.

Figure 34. Setup License Server Installer


pure::variants License Server 5.0 Setup

Welcome to pure::variants License Server 5.0 Setup

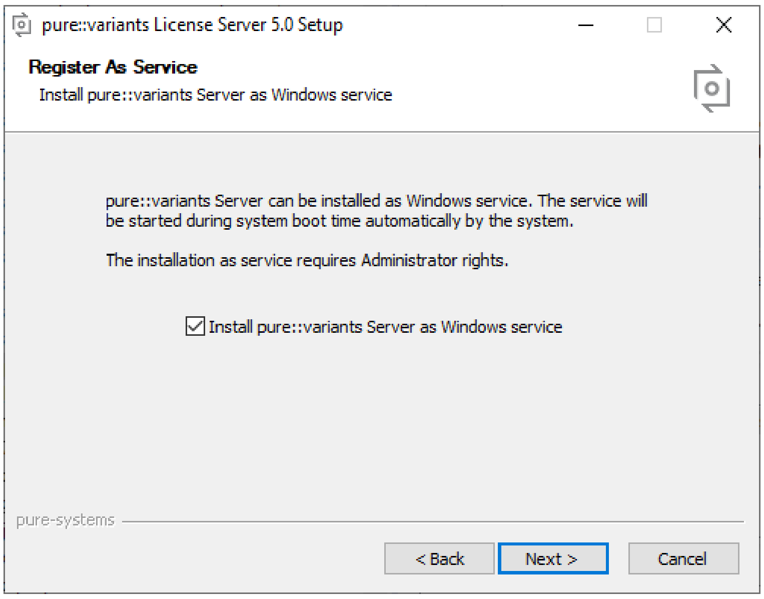
Setup will guide you through the installation of pure::variants License Server 5.0.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Click *Next*.

Figure 35. Setup License Server Windows Service


pure::variants License Server 5.0 Setup

Register As Service

Install pure::variants Server as Windows service

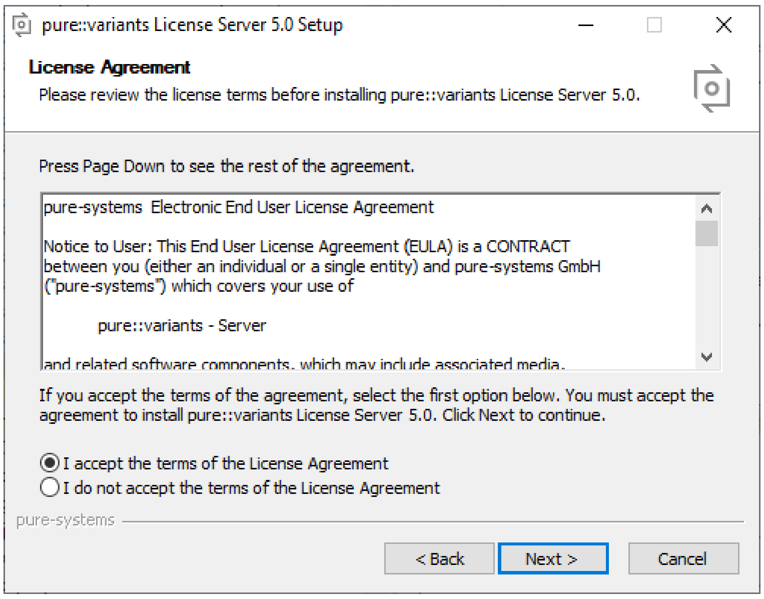
pure::variants Server can be installed as Windows service. The service will be started during system boot time automatically by the system.

The installation as service requires Administrator rights.

[✓] Install pure::variants Server as Windows service

Decide whether the license server should run as Windows service. It is recommended to run the license server as Windows service. This will ensure that the license server will be started automatically during system startup. Click *Next*.

Figure 36. Setup License Server License



pure::variants License Server 5.0 Setup

License Agreement

Please review the license terms before installing pure::variants License Server 5.0.

Press Page Down to see the rest of the agreement.

```
pure-systems  Electronic End User License Agreement

Notice to User: This End User License Agreement (EULA) is a CONTRACT
between you (either an individual or a single entity) and pure-systems GmbH
("pure-systems") which covers your use of

        pure::variants - Server

and related software components, which may include associated media,
```

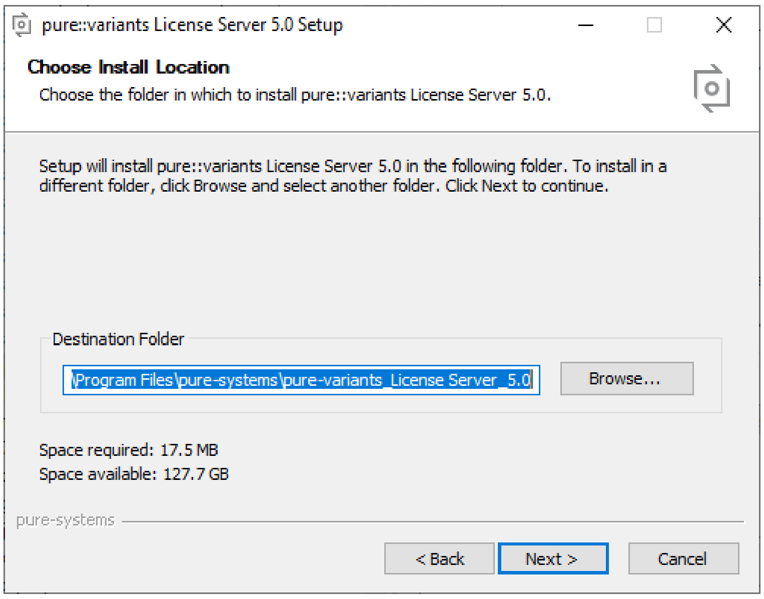
If you accept the terms of the agreement, select the first option below. You must accept the agreement to install pure::variants License Server 5.0. Click Next to continue.

[✓ I accept the terms of the License Agreement]
I do not accept the terms of the License Agreement

pure-systems

Read the license agreement, and after accepting it click *Next*.

Figure 37. Setup License Server Installation Location



pure::variants License Server 5.0 Setup

Choose Install Location

Choose the folder in which to install pure::variants License Server 5.0.

Setup will install pure::variants License Server 5.0 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

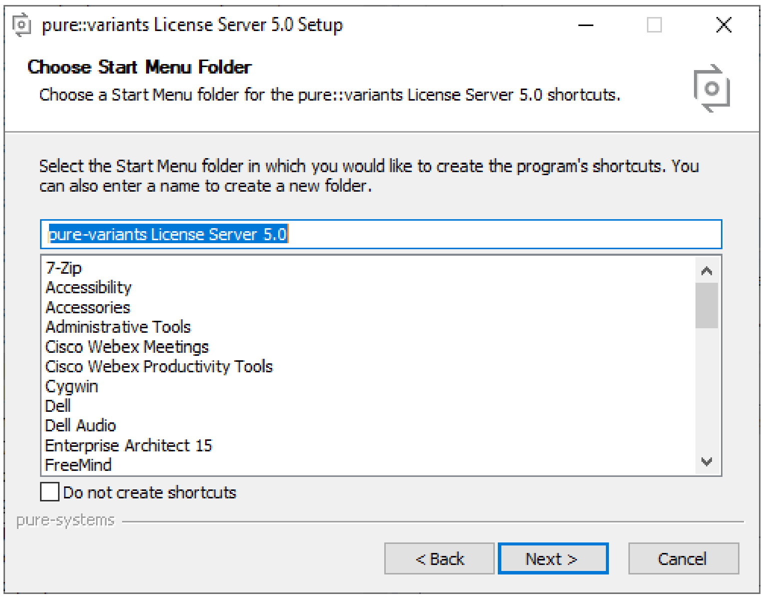
Destination Folder: \Program Files\pure-systems\pure-variants_License Server_5.0

Space required: 17.5 MB
Space available: 127.7 GB

pure-systems

Select the folder where to install the license server files. Click *Next*.

Figure 38. Setup License Server Start Menu


pure::variants License Server 5.0 Setup

Choose Start Menu Folder

Choose a Start Menu folder for the pure::variants License Server 5.0 shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

pure-variants License Server 5.0

7-Zip
Accessibility
Accessories
Administrative Tools
Cisco Webex Meetings
Cisco Webex Productivity Tools
Cygwin
Dell
Dell Audio
Enterprise Architect 15
FreeMind

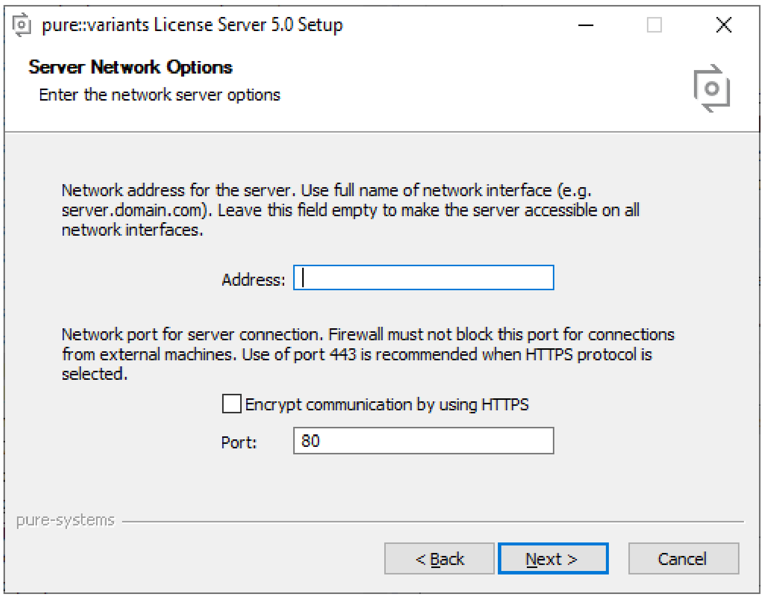
☐ Do not create shortcuts

pure-systems

< Back | Next > | Cancel

Enter a name for the Windows start menu entry, or disable the creation of the start menu entry. Click *Next*.

Figure 39. Setup License Server Network Options


pure::variants License Server 5.0 Setup

Server Network Options

Enter the network server options

Network address for the server. Use full name of network interface (e.g. server.domain.com). Leave this field empty to make the server accessible on all network interfaces.

Address:

Network port for server connection. Firewall must not block this port for connections from external machines. Use of port 443 is recommended when HTTPS protocol is selected.

☐ Encrypt communication by using HTTPS

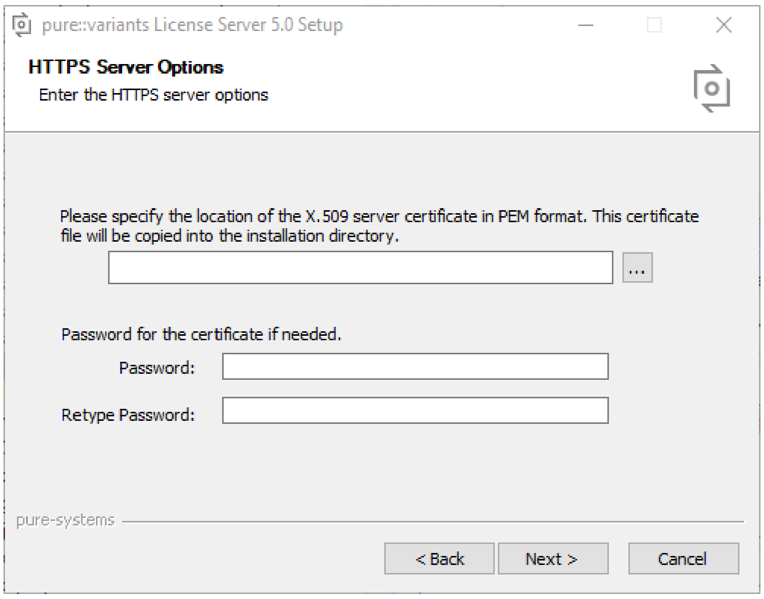
Port: 80

pure-systems

< Back | Next > | Cancel

Now configure the network options. The *Address* field specifies the hostname or IP address on which to make the license service available. If you leave this field empty, the license server will automatically be available on all available network interfaces. You can encrypt the communication with the license server by selecting the *Encrypt* option. This will enable HTTPS instead of HTTP. This requires an X.509 certificate for the license server. The certificate setup is done on a separate page in the installer. The port for communication can be specified in the *Port* field. The default values (80 for HTTP and 443 for HTTPS) should work best. Please ensure that the chosen port is not blocked by a firewall or used by another service. Click *Next*.

Figure 40. Setup License Server Encryption Settings



pure::variants License Server 5.0 Setup

HTTPS Server Options

Enter the HTTPS server options

Please specify the location of the X.509 server certificate in PEM format. This certificate file will be copied into the installation directory.

Password for the certificate if needed.

Password:

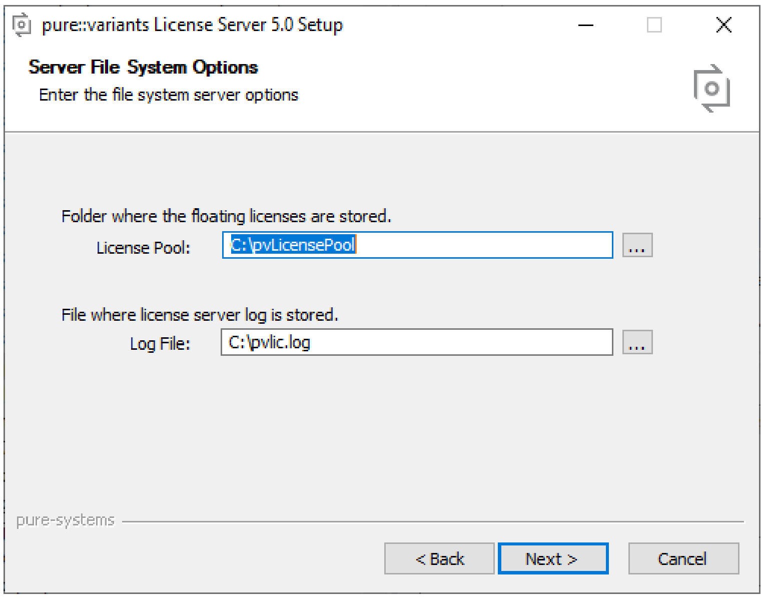
Retype Password:

pure-systems

< Back | Next > | Cancel

If encryption was enabled, enter the path to the X.509 certificate into the upper file field. The format of the certificate needs to be PEM. No other types are supported. If the certificate is protected by a password, enter the password into the two password fields. Click *Next*.

Figure 41. Setup License Server Licenses Options



pure::variants License Server 5.0 Setup

Server File System Options

Enter the file system server options

Folder where the floating licenses are stored.

License Pool: C:\pvLicensePool

File where license server log is stored.

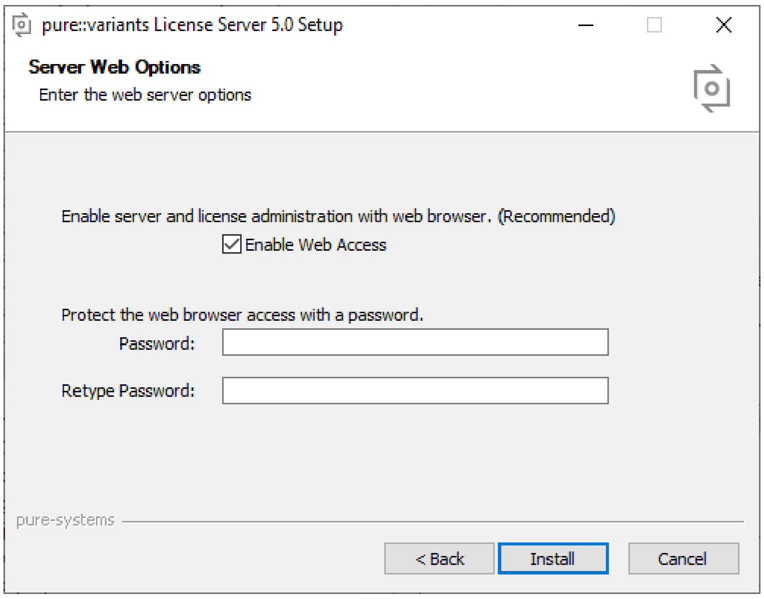
Log File: C:\pvlic.log

pure-systems

< Back | Next > | Cancel

Please select a directory where the license files will be stored. You can place your server license file(s) in this folder now, or use the license server's web interface later to install the licenses. Specify also a location for the server log file. Click *Next*.

Figure 42. Setup License Server Web Options


pure::variants License Server 5.0 Setup

Server Web Options

Enter the web server options

Enable server and license administration with web browser. (Recommended)

☑ Enable Web Access

Protect the web browser access with a password.

Password:

Retype Password:

The license server provides a web interface. The web interface can be enabled or disabled on this page. It allows installing and updating licenses as well as the management of currently used licenses. Please secure the web interface with a password. If you do so, then ensure to also have HTTPS enabled. Otherwise the password will be transmitted not encrypted and could be spied on.

Click *Install* to start the installation process.

If the Windows service option was chosen, the license server will start automatically after successful installation.

4.1.2. Install from Archive

If using the installer is not an option, the server is also distributed as compressed archive file.

Download the license server archive from the pure::variants product web page. Go to http://www.pure-systems.com/pvde-update/.

The product download pages are protected by a password. You need to login by using the email address and registration number from the server license file provided to you by pure-systems.

Create a directory for the pure::variants license server on your disk and extract the contents of the archive into that folder. For Linux systems a recommended location is the directory */opt/pure-variants*. On Windows a recommended location is the directory *C:\Program Files\pure-systems\pure-variants_License_Server_5.0*

Create a directory for the license files. A proper location for Linux systems is */opt/pure-variants/licenses* and on Windows *C:\Program Files\pure-systems\pure-variants_License_Server_5.0\licenses*. Copy the provided server license file into that folder.

Open script *start.bat* on Windows resp. *start.sh* on Linux in a text editor. The script is located in the *server* subdirectory. Update the following variables to match your environment:

HOSTNAME: The hostname or IP address of the server machine the license server should bind to.

PORT: The TCP port to bind to on the server machine.

LICENSEDIR: The folder containing the license files.

Clients must be able to connect to the given port on the server machine. Therefore this port must not be blocked by a firewall or used by another service. A port number can range from 0 to 65535. Port 80 is the standard port for the HTTP and 443 for the HTTPS protocol. Using these standard ports can help with firewall problems.

The server can be started by running script *start.bat* on Windows resp. *start.sh* on Linux. Please consult your system documentation how to add this script to the system's start sequence to ensure the server is started automatically after system reboot. You are now able to connect to the pure::variants License Server with your pure::variants Enterprise clients.

See Section 4.4, “Basic Setup with the pure::variants License Server Web Interface” for further installation steps.

On Windows the pure::variants server can be registered as a service. Selecting this option ensures that the pure::variants server always runs after reboot even when no user is logged in. To register the pure::variants server as a service, use the command line option /install. If installing the pure::variants license server as a service, a service name has to be given with command line option /servicename. A service description can be added with command line option /servicedesc. Here is an example how to register the license server as service:

```
<license server install path>/server/variantsd.exe /install
/servicename "pure--variants License Server" /servicedesc "The license server is providing
licenses for the pure::variants client"
```

Enable pure::variants License Server Web Interface

The pure::variants license server includes an optional web interface for monitoring and interacting with the server.

To enable the server's web interface on Windows, add the following options to the server command line at the end of the script *start.bat*:

```
/enableweb
/webpwd PASSWORD
```

To enable the server's web interface on Linux, add the following options to the server command line at the end of the script *start.sh*:

```
--enableweb
--webpwd PASSWORD
```

The server's web interface should be always protected by a password. Please use a secure password for this.

4.2. Update pure::variants License Server

4.2.1. Update with Windows Installer (Windows Only)

Updating an existing pure::variants license server works exactly the same as a clean installation of the pure::variants license server. Please see Section 4.1.1, “Install with Windows Installer (Windows only)”. Additional settings performed in the server configuration file will remain in the server configuration file after updating.

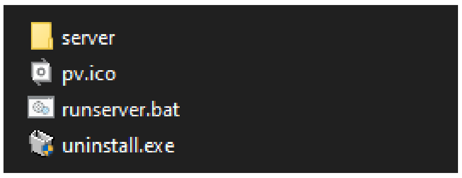
4.2.2. Update with Archive

Updating an existing pure::variants license server works exactly the same as a clean installation of the pure::variants license server. Please see Section 4.1.2, “Install from Archive”.

4.3. Uninstall pure::variants License Server

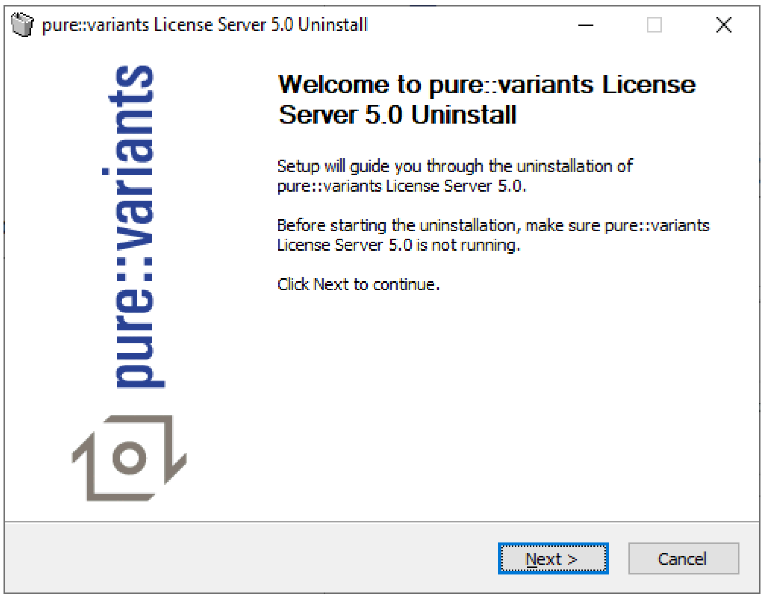
The uninstaller for the pure::variants license server can be started in two different ways. The first one is to go to the Windows *Add or remove programs* application and search for *pure::variants License Server 5.0* and start the uninstaller by using the *Uninstall* action. The uninstaller requires Administrator privileges.

Figure 43. pure::variants License Server Uninstaller



The second possibility is to navigate to the pure::variants license server installation folder and start the uninstaller by double-clicking it.

Figure 44. pure::variants License Server Uninstaller



pure::variants License Server 5.0 Uninstall

Welcome to pure::variants License Server 5.0 Uninstall

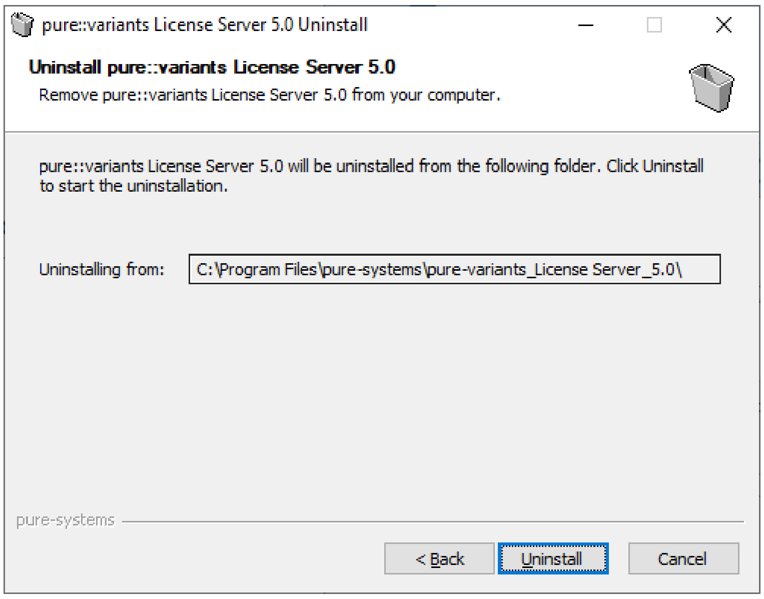
Setup will guide you through the uninstallation of pure::variants License Server 5.0.

Before starting the uninstallation, make sure pure::variants License Server 5.0 is not running.

Click Next to continue.

Click *Next*.

Figure 45. Uninstall from



pure::variants License Server 5.0 Uninstall

Uninstall pure::variants License Server 5.0

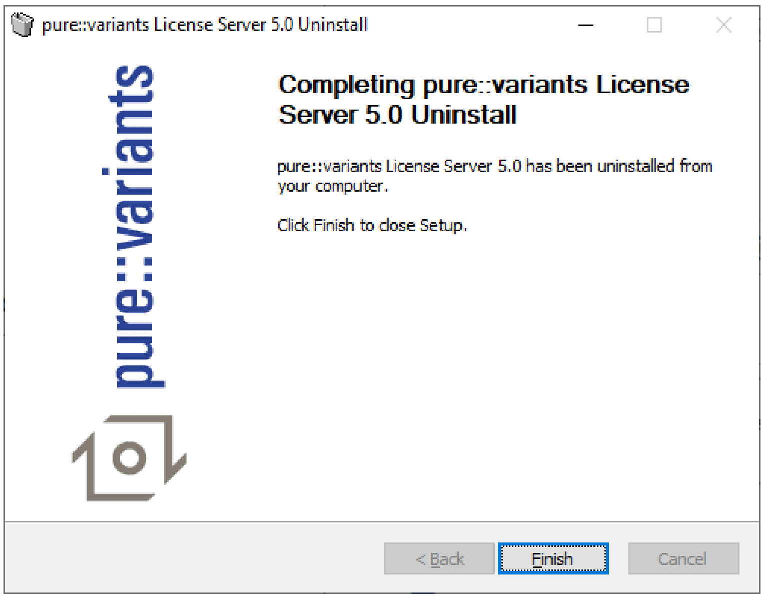
Remove pure::variants License Server 5.0 from your computer.

pure::variants License Server 5.0 will be uninstalled from the following folder. Click Uninstall to start the uninstallation.

Uninstalling from: C:\Program Files\pure-systems\pure-variants_License Server_5.0\

Click *Uninstall* to start the uninstall process.

Figure 46. Completing Uninstall


pure::variants License Server 5.0 Uninstall

Completing pure::variants License Server 5.0 Uninstall

pure::variants License Server 5.0 has been uninstalled from your computer.

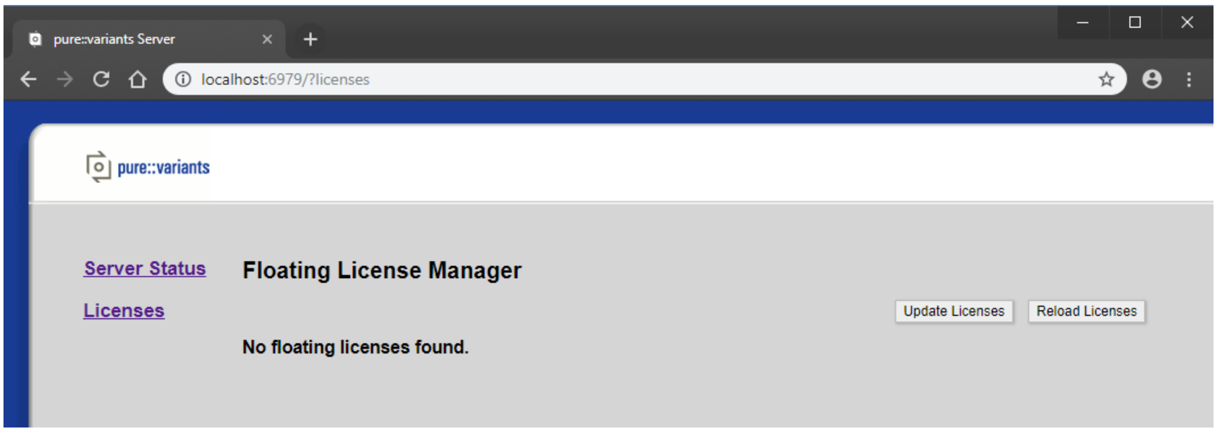
Click Finish to close Setup.

The deinstallation is succesfully finished. Click *Finish* to close the uninstaller.

4.4. Basic Setup with the pure::variants License Server Web Interface

To add your floating licenses to the server, open the web interface. If a password was enabled during installation, the web interface will show a *Login* button. Click the button, enter the password, and leave the user field empty. Go to the *"License"* page.

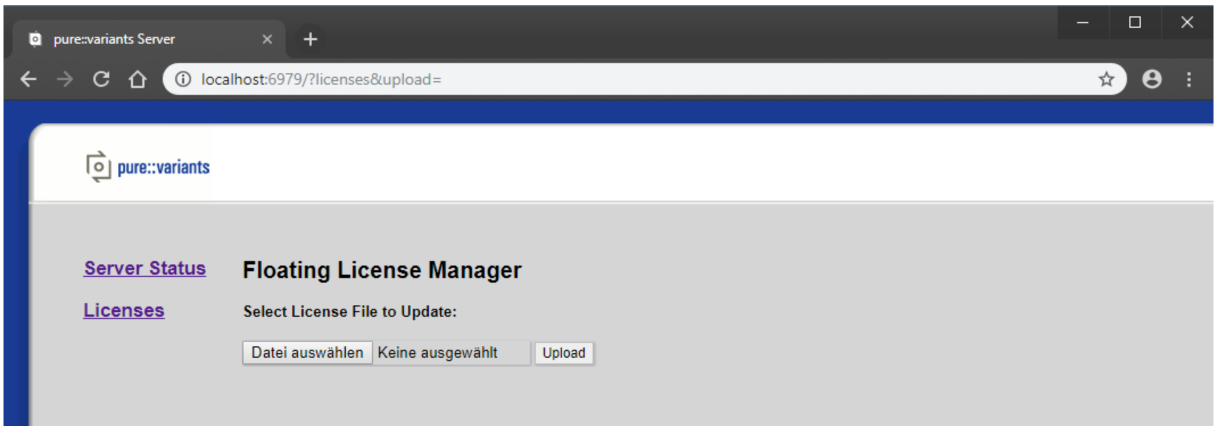
Figure 47. License Server Web Interface


Floating License Manager

No floating licenses found.

Because there is no license installed, the web page shows a *No floating licenses found* message. Click *Update Licenses*.

Figure 48. Upload License

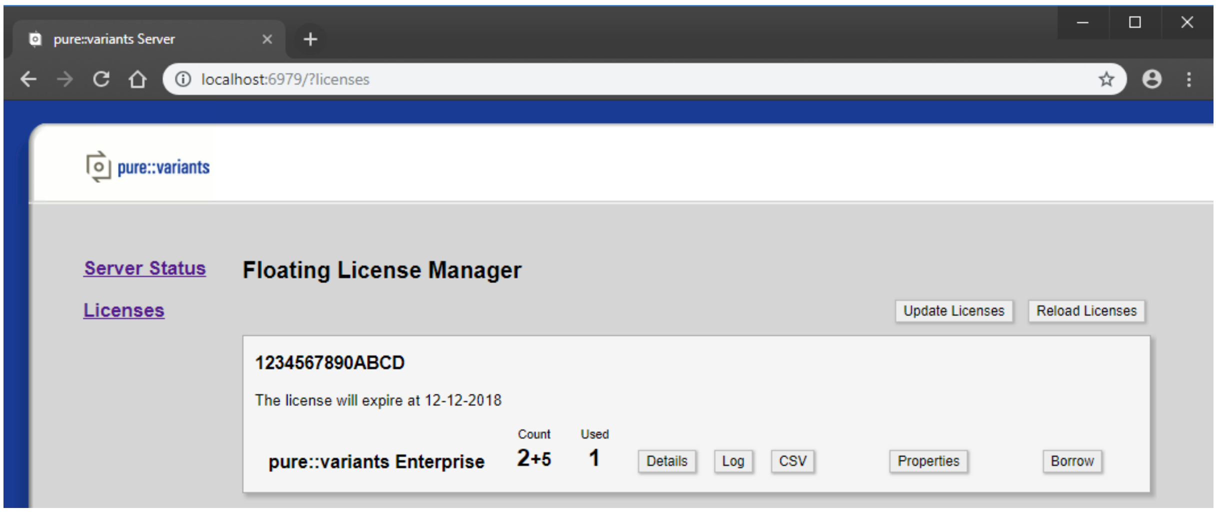


Floating License Manager

Select License File to Update:

Select the server license file from your hard disk. Click *Upload*.

Figure 49. License Overview



Floating License Manager

1234567890ABCD

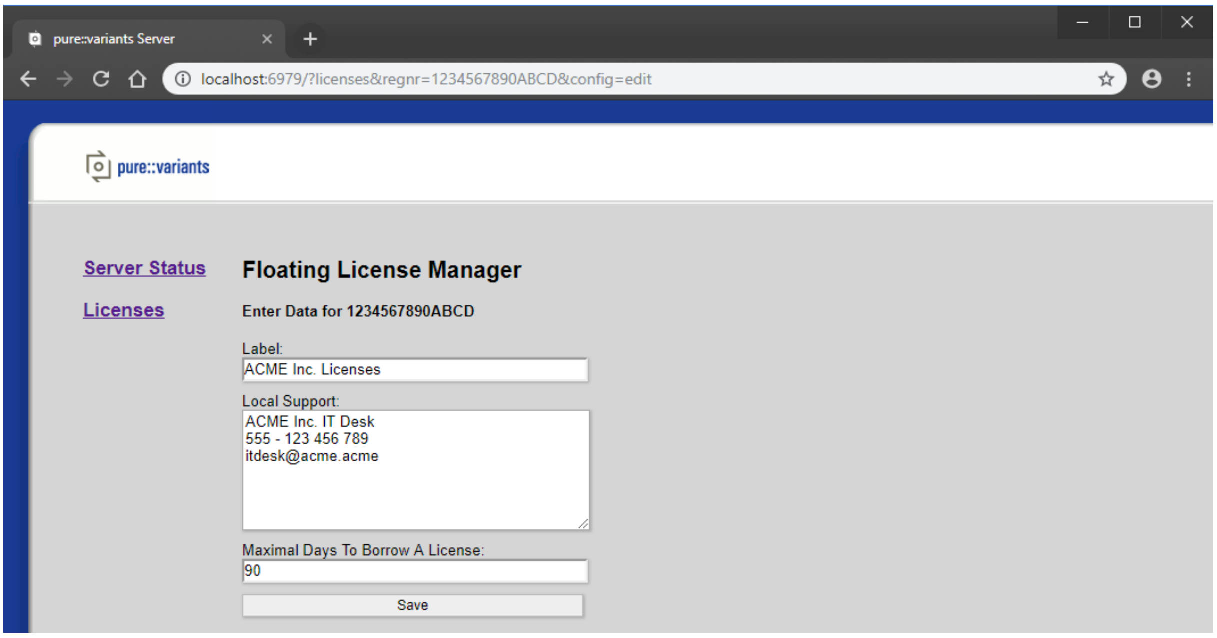
The license will expire at 12-12-2018

	Count	Used
pure::variants Enterprise	2+5	1

After successful license update, the license entry is shown on the web page. The entry shows your registration number and the number of available and used licenses. The server is able to serve multiple licenses. The different licenses will be distinguished by the registration number.

For easier usage every license entry can get a label. Click on *Properties* to enter a label and local support information for the license.

Figure 50. License Server Detailed License Data



Floating License Manager

Enter Data for 1234567890ABCD

Label:
ACME Inc. Licenses

Local Support:
ACME Inc. IT Desk
555 - 123 456 789
itdesk@acme.acme

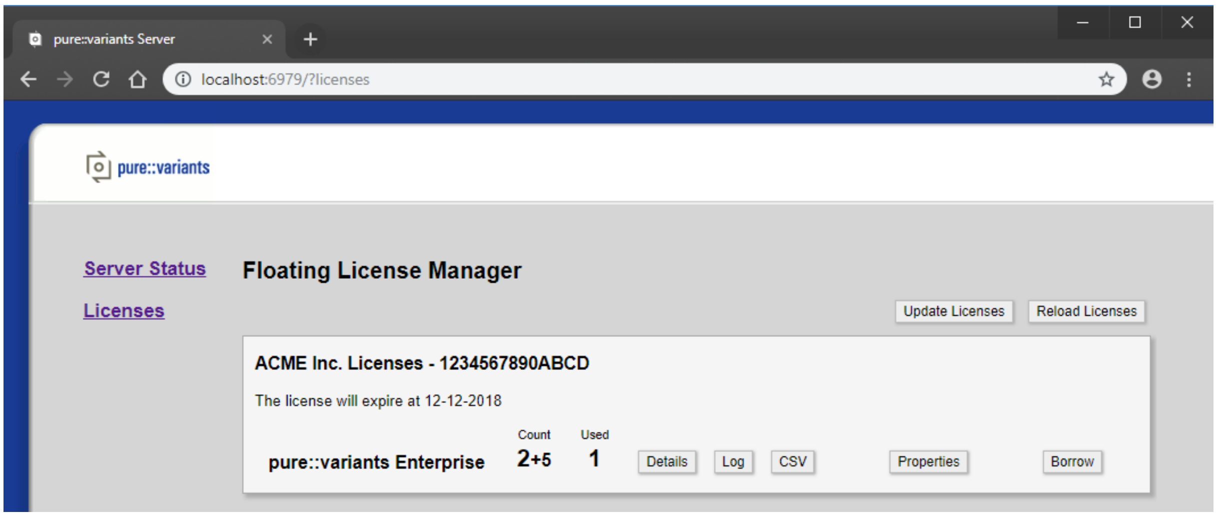
Maximal Days To Borrow A License:
90

The label is shown on the web page in front of the registration number. The local support data is presented to the clients in case of license access errors. Enter your IT help desk contact data here to inform the user who can be called in case of any problems.

The *Maximal Days To Borrow A License* property defines the maximum amount of days a license can be borrowed by the client or in the web interface. If the value is set to 0, borrowing of licenses is disabled. pure::variants clients can use a maximum amount of 90 days for borrowing a license. Longer time intervals can be used in the web interface only.

Click *Save* to store the data.

Figure 51. License Overview



Floating License Manager

ACME Inc. Licenses - 1234567890ABCD

The license will expire at 12-12-2018

	Count	Used
pure::variants Enterprise	2+5	1

Now the entered label is shown in the license entry.

4.5. License Server Command Line Options

For generic pure::variants server command line options see Section 5.7, “Server Command Line Options”.

Following list describes the license server specific options that can be added to the pure::variants license server command line or configuration file.

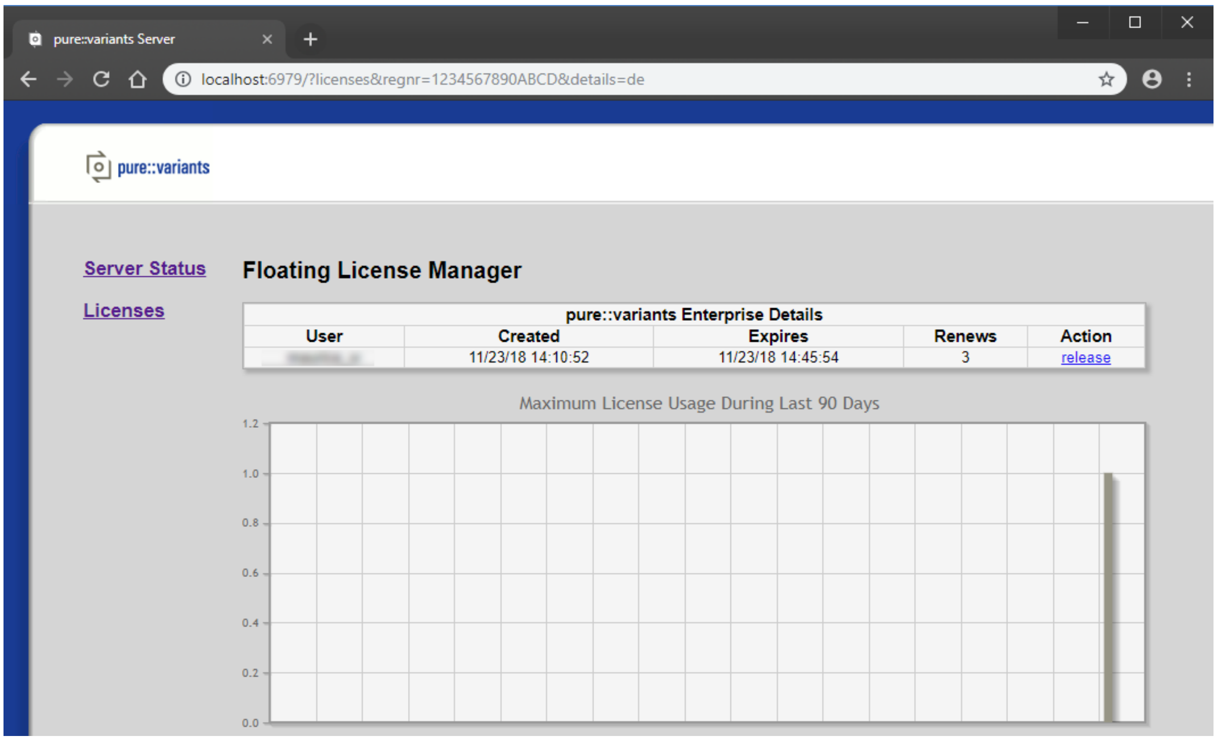
See Section 5.4, “Location of the Server Configuration File” to find the configuration file.

Option	Description
/license	Location of the directory containing the license files, or the location of a single license file. This parameter can be specified multiple times.
/licenselog	Enable license usage logging
/licenseuserlog	Enable license usage logging including user data
/licenseuserlist [PATH]	Location of a user license access control list.

4.6. Get information about the license usage

For every license entry you can view the current usage details and a chronological log by clicking the *Details* or *Log* button. After pressing the *CSV* button the license log of the last 90 days is exported into a CSV file.

Figure 52. License Details


pure::variants Server
localhost:6979/?licenses&regnr=1234567890ABCD&details=de
pure::variants
Server Status
Licenses

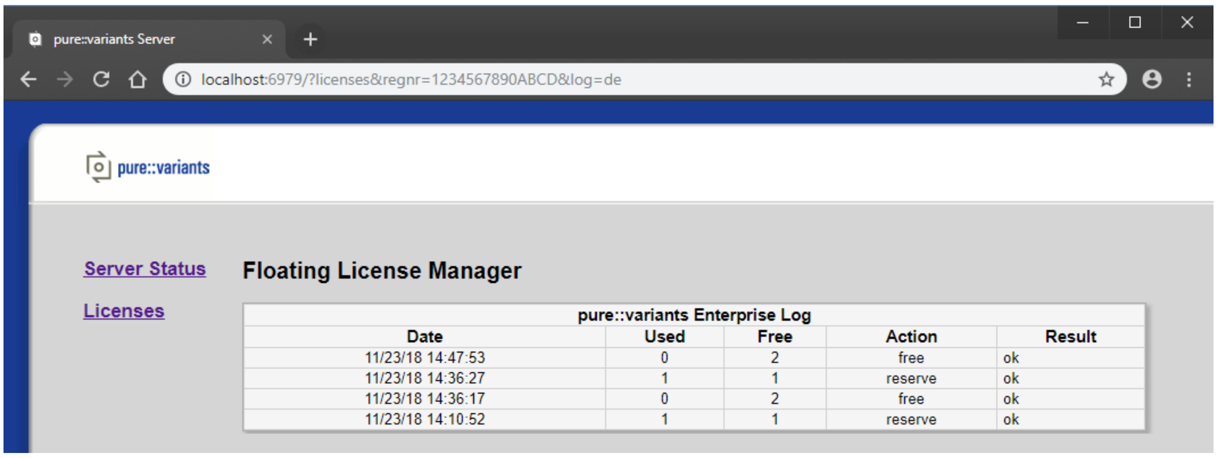
Floating License Manager

pure::variants Enterprise Details				
User	Created	Expires	Renews	Action
[redacted]	11/23/18 14:10:52	11/23/18 14:45:54	3	release

Maximum License Usage During Last 90 Days
1.2
1.0
0.8
0.6
0.4
0.2
0.0

Pressing *Details* opens the license details page. The upper part of this page shows who is currently using a license. The lower part shows the maximal license usage during the last 90 days.

Figure 53. License Log



Floating License Manager

pure::variants Enterprise Log

Date	Used	Free	Action	Result
11/23/18 14:47:53	0	2	free	ok
11/23/18 14:36:27	1	1	reserve	ok
11/23/18 14:36:17	0	2	free	ok
11/23/18 14:10:52	1	1	reserve	ok

The table on the log page shows all license reserve and free operations. Possible errors are also shown here.

4.7. Borrow a license in the web interface

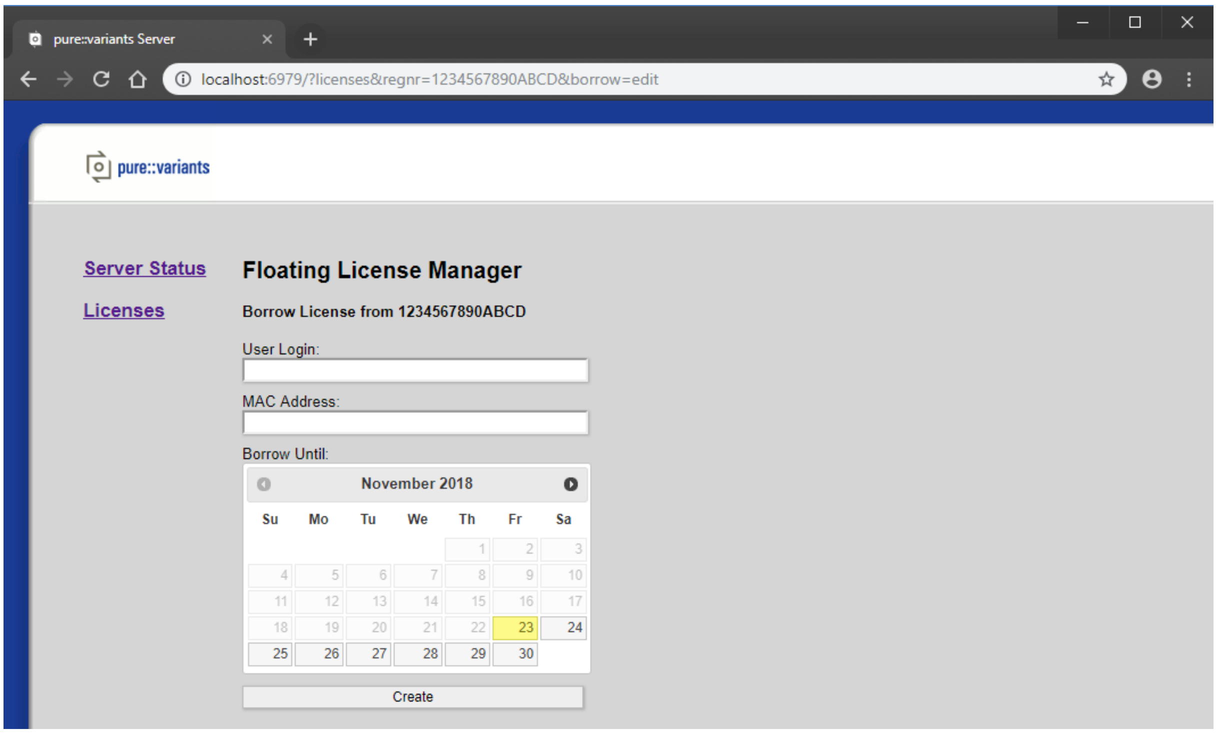
The web interface provides the possibility to reserve an offline license. This offline license can be used like a normal client license.

With the *Borrow* button on one of the license entries it is possible to create a client license for a specified amount of days. On the next page the user login name, the MAC address of the client computer, and the expiration date need to be specified.

Be careful, borrowing a license will block the license until the specified expiration date is reached. This can not be reverted. So check the defined login name and MAC address properly before creating the license.

After using the *Create* button, the client license is created and downloaded automatically.

Figure 54. Borrow a License



Floating License Manager

Borrow License from 1234567890ABCD

User Login:

MAC Address:

Borrow Until:

4.8. Adding the License Server Information to the Client License

The client license file may be optionally changed to contain the information where the pure::variants license server is located.

The *url* attribute of the *subtype* tag is initially empty and may contain the server URL constructed from the information given during the installation process.

Note

Always use the full domain name including the port number (even for standard ports 80 and 443).

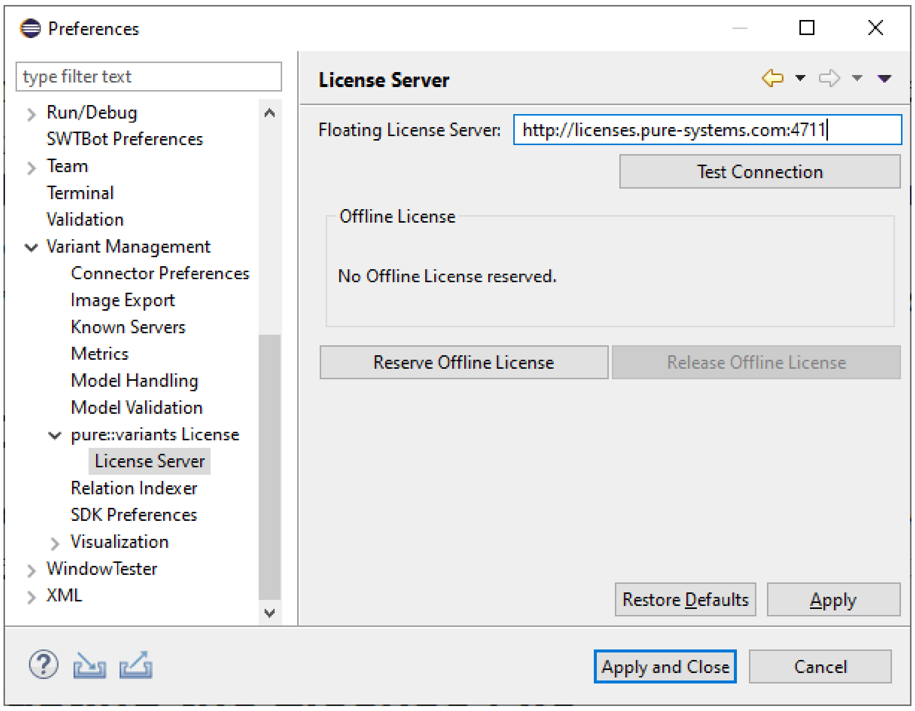
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<licence version="1.1">
  <product>
    <name>pure::variants</name>
    <version>2</version>
    <function>de</function>
    <platform>win32</platform>
    <platform>linux</platform>
    <platform>macos</platform>
  </product>
  <creation>1-1-2000</creation>
  <type>float</type>
  <subtype url="http://pvlicense.example.com:80">licence server</subtype>
  <registration>
    <firma>ACME Inc.</firma>
    <name>Jane Doe</name>
    <email>jane.doe@acme.acme</email>
    <number>1234567890ABCD</number>
  </registration>

<signature>12..EF</signature></licence>
```

4.8.1. Setup License Server Location

This step is necessary only if a floating license with a pure::variants license server is used. The license server URL can be provided with the floating license file, or it has to be set by the user. This setting has to be performed once per workspace in the pure::variants client.

To set the license server URL, open the **Preferences** (menu *Window* -> *Preferences*). Navigate to **Variant Management -> pure::variants License -> License Server** and enter the URL of the license server into the **Floating License Server** text field.

Figure 55. pure::variants License Server Preferences



Floating License Server: http://licenses.pure-systems.com:4711

Offline License

No Offline License reserved.

4.9. Add a license access control list

With a license access control list it can be controlled which users are allowed to get a license from the pure::variants License Server, globally in Single License Mode as well as for specific license registration numbers in Multi License Mode.

Users can be allowed and denied to get a license separately. A license is granted to a user by the pure::variants License Server in Single License Mode if

- no license access control list is configured, or
- a global license access control list is configured, and
  - the user is explicitly allowed, or no user is explicitly allowed, and
  - the user is not explicitly denied

In Multi License Mode a license is granted to a user by the pure::variants License Server if

- no license access control list is configured for the license registration number for which the user requests a license, or
- a license access control list is configured for the license registration number for which the user requests a license, and
  - the user is explicitly allowed, or no user is explicitly allowed, and
  - the user is not explicitly denied

A license access control list is a text file with the following (XML) format.

```
<userlists ignorecase="true">

  <!-- global access control list -->
  <userlist>
```

```
    <allow>username1</allow>
    <allow>username2</allow>
    ...
    <deny>username3</deny>
    <deny>username4</deny>
    ...
  </userlist>

  <!-- for registration number XYZABC123 only -->
  <userlist regnr="XYZABC123">
    <allow>username1</allow>
    <allow>username2</allow>
    ...
    <deny>username3</deny>
    <deny>username4</deny>
    ...
  </userlist>

  ...
</userlists>
```

The listed user names are case sensitive. This can be changed by setting the optional attribute *ignorecase* to the value *true*.

The following example only allows “user1” to get a license for the license registration number “ABC”. No other user can get a license for this registration number.

```
<userlists ignorecase="false">
  <userlist regnr="ABC">
    <allow>user1</allow>
  </userlist>
</userlists>
```

The next example allows all users except of “user1” to get a license for the registration number “ABC”.

```
<userlists>
  <userlist regnr="ABC">
    <deny>user1</deny>
  </userlist>
</userlists>
```

And the last example allows the users “user1” and “user2” to get a license for the registration number “ABC”, and all users except of users “user3” and “user4” to get a license for the registration number “123”.

```
<userlists>
  <userlist regnr="ABC">
    <allow>user1</allow>
    <allow>user2</allow>
  </userlist>
  <userlist regnr="123">
    <deny>user3</deny>
    <deny>user4</deny>
  </userlist>
</userlists>
```

The license access control list file is considered by the pure::variants License Server by adding command line option *licenseuserlist* to the server configuration file or server command line. This option expects the path to the license access control file as argument. Example:

```
/licenseuserlist “C:\pvLicenseACL.xml”
```

See Section 5.4, “Location of the Server Configuration File” to find the configuration file.

Just add the option as new line to the configuration file and either restart the pure::variants License Server, or click the *Reload Licenses* button on the *Licenses* page of the pure::variants License Server Web Interface. This can also be automated by opening the Web interface page with the URL parameter “reload=” added.

Examples for the tools curl and wget:

- curl "http://server:port/?licenses&reload="
- wget -O- "http://server:port/?licenses&reload="

If a password for the web interface is defined, the requests have to provide this password as well. The user has to be specified since the tools are not working without. Just leave “admin” here. The “password” has to be replaced by your password.

- curl "http://server:port/?licenses&reload=" -uadmin:password
- wget -O- "http://server:port/?licenses&reload=" --user=admin --password=password

4.10. Proxy Configuration

The pure::variants license server can be placed behind a reverse proxy. No configuration changes are required on the side of the license server.

Since the license server only uses relative links, e.g. to load image, CSS, and JavaScript files for its web interface from the server, the reverse proxy needs to correct the paths in the HTTP response headers (redirect).

The following code shows a minimal example for an Nginx reverse proxy configuration suitable for the pure::variants license server.

```
events {
  worker_connections 1024;
}

http {
  server {
    listen 80;
    server_name proxy;

    location /pvlic/ {
      proxy_pass     http://backend:8081/;
      proxy_redirect http://backend:8081 /pvlic;
    }
  }
}
```

This configuration passes requests addressed to *http://proxy/pvlic/* to the backend server address *http://backend:8081/*. The server names *proxy* and *backend* only are examples and need to be replaced with your actual server names. The same applies to the ports and the location path in the example.

The following code shows the same example for Apache and IBM HTTP Server (just the proxy part).

```
ProxyPass        /pvlic/ http://backend:8081/
ProxyPassReverse /pvlic  http://backend:8081
```

5. pure::variants Model Server

5.1. Install pure::variants Model Server

5.1.1. pure::variants Database Model Server Installation

The installation of the pure::variants Database Model Server requires 3 steps:

- Setup of the database and the ODBC data source
- Installation of the pure::variants Server on the server host
- Basic setup of the server

There are two alternative ways to install pure::variants, depending on your operating system. The different installation procedures are described below.

Upgrading from earlier versions prior to 5.0.3

If you already deployed the pure::variants Server and migrate to the pure::variants Server 5.0.3, updating the database is recommended to benefit from an optimized database layout.

Setup of the database and the ODBC data source

Before installing the pure::variants server, a database has to be created. Please read the documentation of your database system how to setup a new database.

For **Oracle**, the new database has to use the **AL32UTF8 character set** to ensure correct handling of multi-lingual content.

As the next step, the fresh database needs to be initialized. For this, execute the provided initialization script for your database system. The script will create the needed table structure, indices, and procedures. The script can be found in the *pure::variants Windows Installer Package*, which can be downloaded from our pure::variants update site. The script is named *Init<DatabaseName>.sql*.

After initialization of the database, create a user and grant privileges to the database. The user needs rights to execute SELECT, INSERT, UPDATE, DELETE, and PROCEDURES. Please consult the documentation of your database system how to perform this.

On the host which will run the pure::variants server, the client software for your database has to be installed and an ODBC data source in the Windows Management Console needs to be setup. Please consult the documentation of your database system how to perform this. Ensure to enable the **Query Timeout** option if available.

Update database layout and data for databases created before pure::variants 5.0.3

The release of pure::variants 5.0.3 contains optimized access to the models stored into a database. To enable this improvements for existing installations the database table layout and stored procedures needs to be updated (Step 1). After this the stored data has to be migrated into the new tables (Step 2). This has to be executed once only. It creates a database layout which is incompatible with earlier versions of pure::variants Server. Older pure::variants Server will not be able to start with a migrated database and cause an error message.

ATTENTION: Before executing the following steps, make sure you have an up-to-date backup of your database. Shutdown the pure::variants Server while executing the migration scripts.

Two SQL scripts are provided for the update of your database. The scripts can be found in the “pure::variants Windows Installer Package”, which can be downloaded from our pure::variants update site. The scripts have to executed by the database administrator which sufficient rights.

1. Update_1_<DatabaseProductName>.sql: The update script creates new tables and replaces stored procedures. After running this script the read and write performance of the pure::variants Server will be increased. The improvement is only available for newly stored data. This is a quick operation.

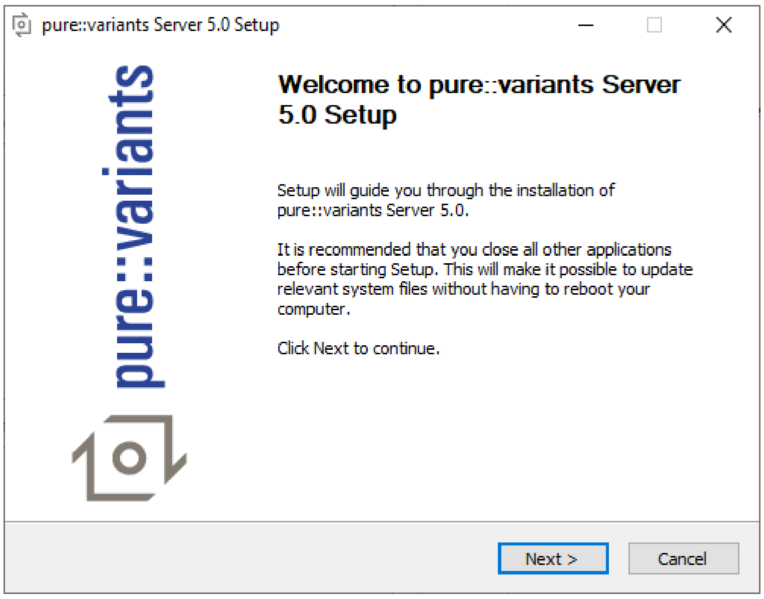
2. Migrate_1_<DatabaseProductName>.sql: The migration script reorganizes the existing data into the newly created tables. After running this script the reading of previously existing data is also improved. This may take a longer time depending on the amount of data in your database.

Install with Windows Installer (Windows Only)

The Windows Installer can be downloaded from the pure::variants product web page. Go to http://www.pure-systems.com/pvde-update. The product download pages are protected by a password. You need to login by using the email address and the registration number from the license file.

Download the model server installer *Setup Database Server X.Y.ZZ.exe* and start the installation by double-clicking the executable. Running the model server installer requires Administrator privileges.

Figure 56. Setup Model Server Installer



Welcome to pure::variants Server 5.0 Setup

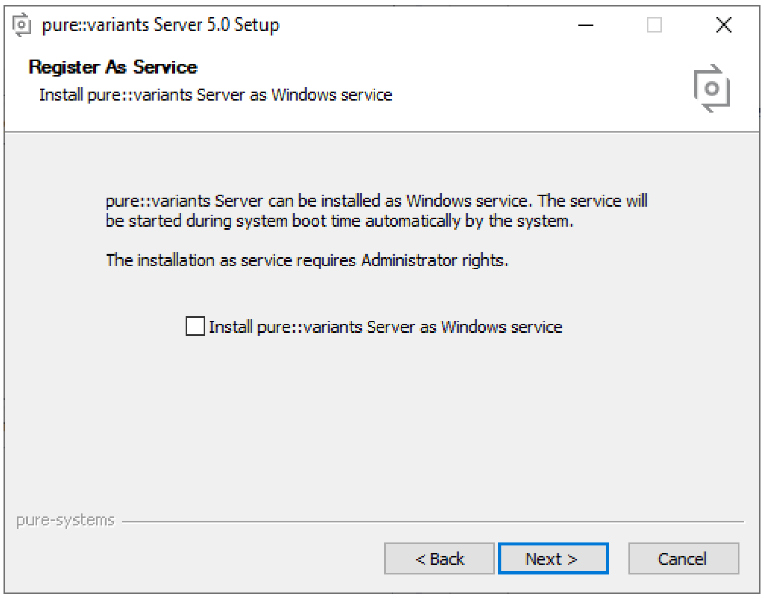
Setup will guide you through the installation of pure::variants Server 5.0.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Click *Next*.

Figure 57. Setup Model Server Windows Service



Register As Service

Install pure::variants Server as Windows service

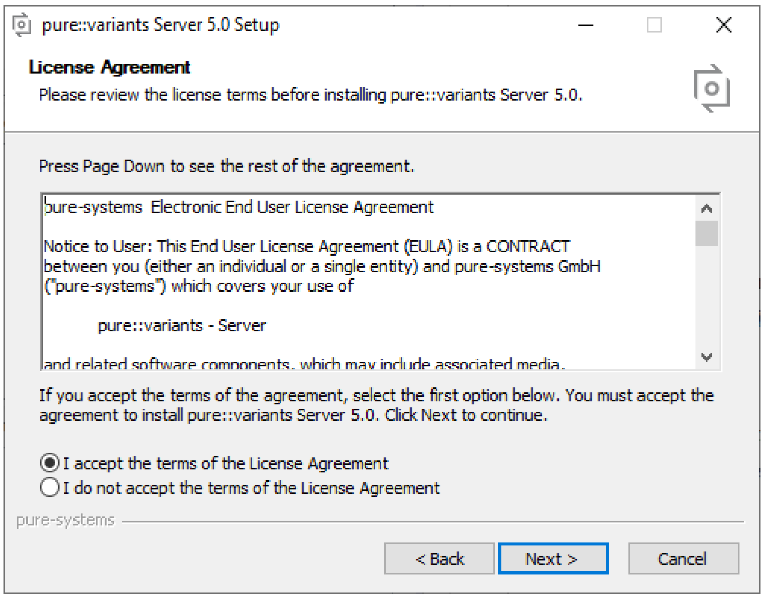
pure::variants Server can be installed as Windows service. The service will be started during system boot time automatically by the system.

The installation as service requires Administrator rights.

☐ Install pure::variants Server as Windows service

Decide whether the model server should run as Windows service. It is recommended to run the model server as Windows service. This will ensure that the model server will be started automatically during system startup. Click *Next*.

Figure 58. Setup Model Server License


pure::variants Server 5.0 Setup

License Agreement

Please review the license terms before installing pure::variants Server 5.0.

Press Page Down to see the rest of the agreement.

pure-systems Electronic End User License Agreement

Notice to User: This End User License Agreement (EULA) is a CONTRACT between you (either an individual or a single entity) and pure-systems GmbH ("pure-systems") which covers your use of

pure::variants - Server

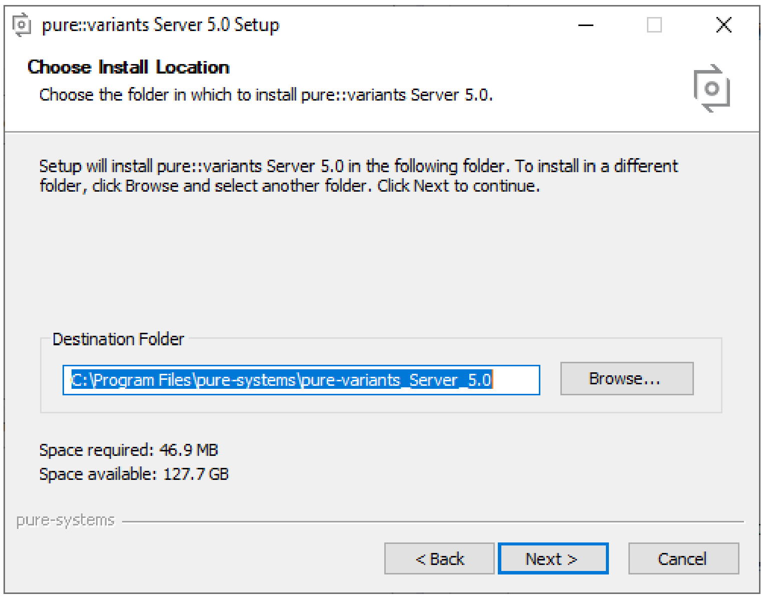
and related software components, which may include associated media,

If you accept the terms of the agreement, select the first option below. You must accept the agreement to install pure::variants Server 5.0. Click Next to continue.

[✓ I accept the terms of the License Agreement]
I do not accept the terms of the License Agreement

Read the license agreement, and after accepting it click *Next*.

Figure 59. Setup Model Server Installation Location


pure::variants Server 5.0 Setup

Choose Install Location

Choose the folder in which to install pure::variants Server 5.0.

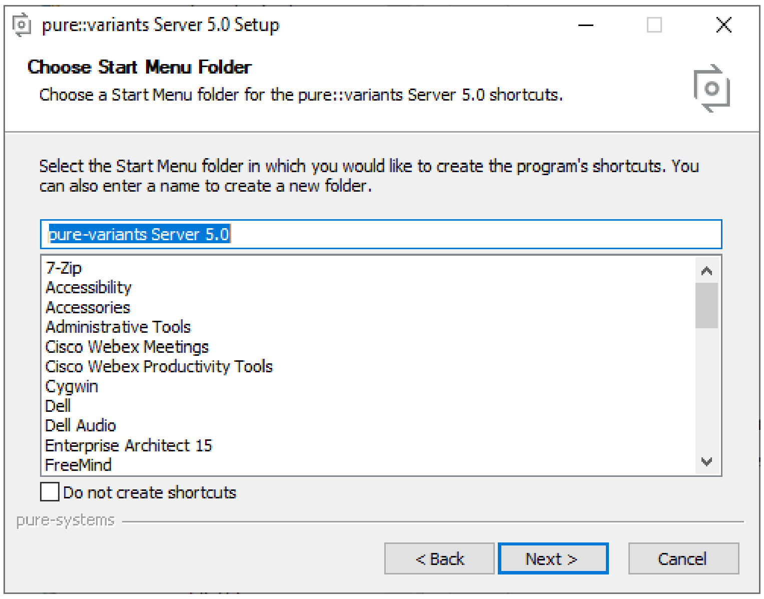
Setup will install pure::variants Server 5.0 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\Program Files\pure-systems\pure-variants_Server_5.0

Space required: 46.9 MB
Space available: 127.7 GB

Select the folder where to install the model server files. Click *Next*.

Figure 60. Setup Model Server Start Menu


pure::variants Server 5.0 Setup

Choose Start Menu Folder

Choose a Start Menu folder for the pure::variants Server 5.0 shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

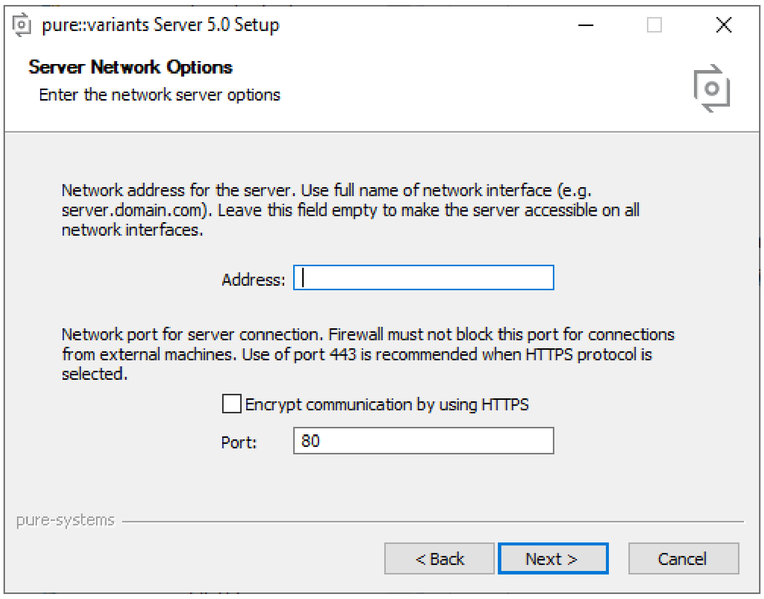
pure-variants Server 5.0

7-Zip
Accessibility
Accessories
Administrative Tools
Cisco Webex Meetings
Cisco Webex Productivity Tools
Cygwin
Dell
Dell Audio
Enterprise Architect 15
FreeMind

☐ Do not create shortcuts

Enter the name of the Windows start menu entry, or disable the creation of the start menu entry. Click *Next*.

Figure 61. Setup Model Server Network Options


pure::variants Server 5.0 Setup

Server Network Options

Enter the network server options

Network address for the server. Use full name of network interface (e.g. server.domain.com). Leave this field empty to make the server accessible on all network interfaces.

Address:

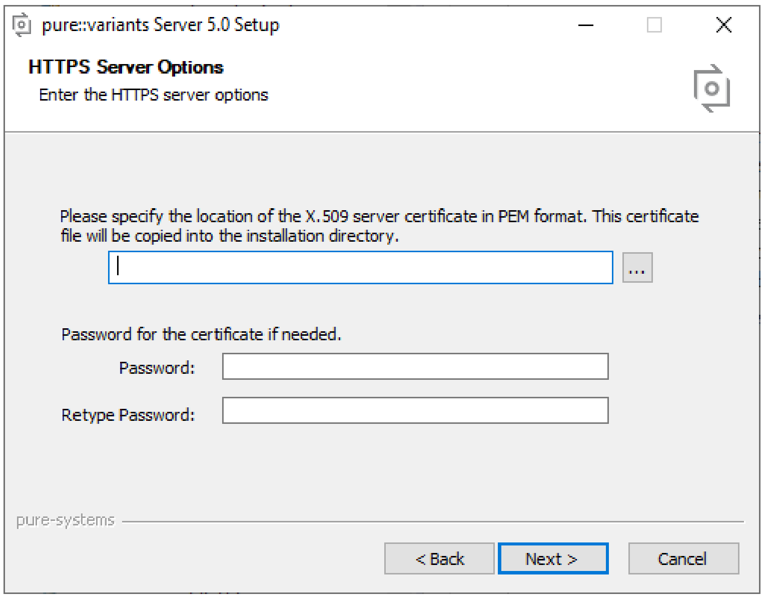
Network port for server connection. Firewall must not block this port for connections from external machines. Use of port 443 is recommended when HTTPS protocol is selected.

☐ Encrypt communication by using HTTPS

Port: 80

Now configure the network options. The *Address* field specifies the hostname or IP address of the network interface on which the model service should run. If you leave this field empty, the model server will automatically work on all available network interfaces. You can encrypt the whole communication by selecting the *Encrypt communication by using HTTPS* option. This will use HTTPS instead of HTTP. This requires an X.509 certificate for the model server. The certificate setup is done on a separate page in the installer. The port for communication can be specified in the *Port* field. The default values (80 for HTTP and 443 for HTTPS) should work best. Please ensure that this port is not blocked by a firewall or used by another service. Click *Next*.

Figure 62. Setup Model Server Encryption


pure::variants Server 5.0 Setup

HTTPS Server Options

Enter the HTTPS server options

Please specify the location of the X.509 server certificate in PEM format. This certificate file will be copied into the installation directory.

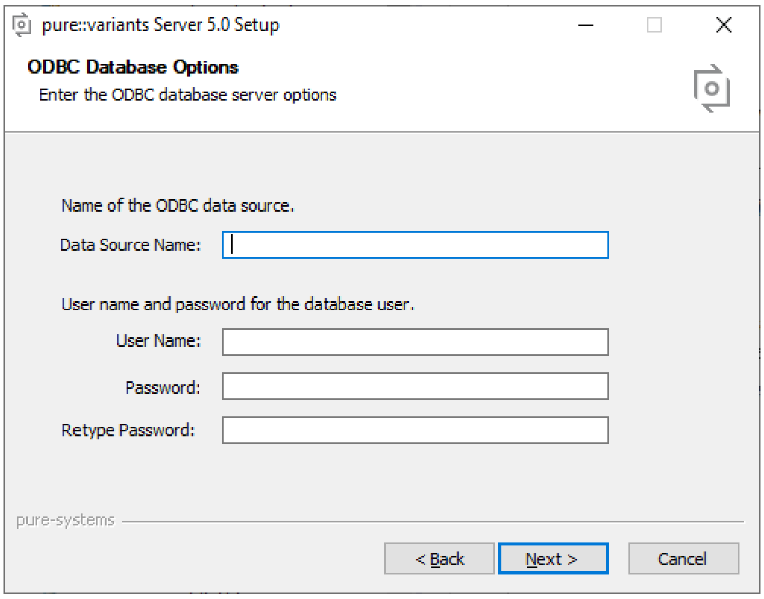
Password for the certificate if needed.

Password:

Retype Password:

If encryption was enabled, enter the path to the X.509 certificate into the upper file field. The format of the certificate needs to be PEM. No other types are supported. If the certificate is protected by a password, enter the password into the two password fields. Click *Next*.

Figure 63. Setup Model Server ODBC Data Source Settings


pure::variants Server 5.0 Setup

ODBC Database Options

Enter the ODBC database server options

Name of the ODBC data source.

Data Source Name:

User name and password for the database user.

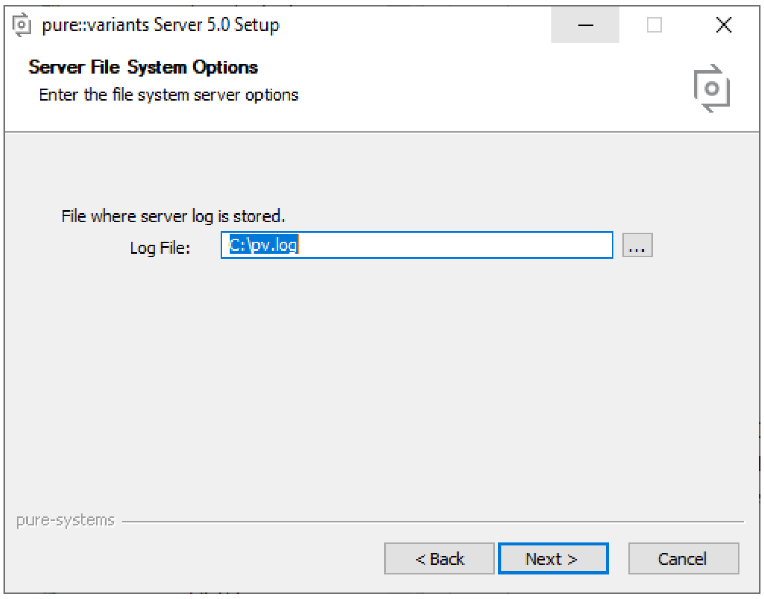
User Name:

Password:

Retype Password:

Enter the ODBC data source name which was created for pure::variants to communicate with the database. To access the database, pure::variants needs the credentials of a user with sufficient access to the database. Click *Next*.

Figure 64. Setup Model Server Log



Server File System Options

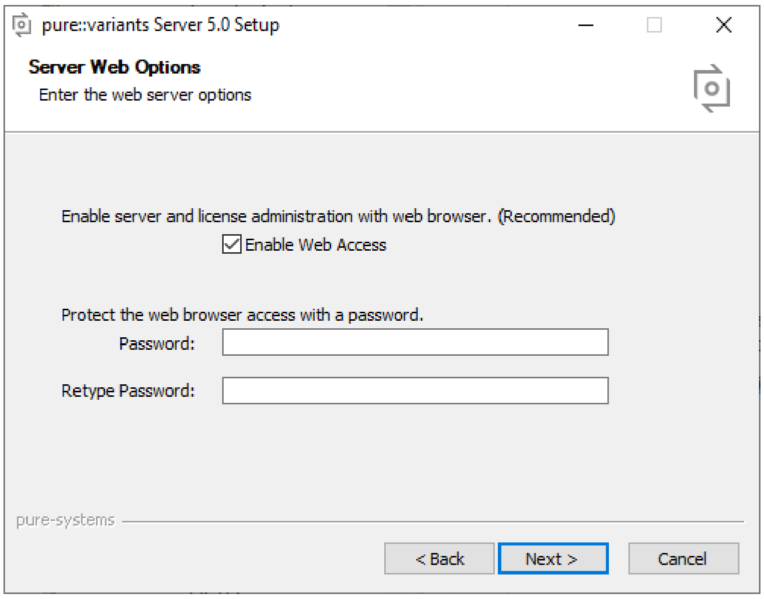
Enter the file system server options

File where server log is stored.

Log File: C:\pv.log

Enter a location for the server log file. The pure::variants server needs to have write access to the specified file. Click *Next*.

Figure 65. Setup Model Server Web Interface



Server Web Options

Enter the web server options

Enable server and license administration with web browser. (Recommended)

☑ Enable Web Access

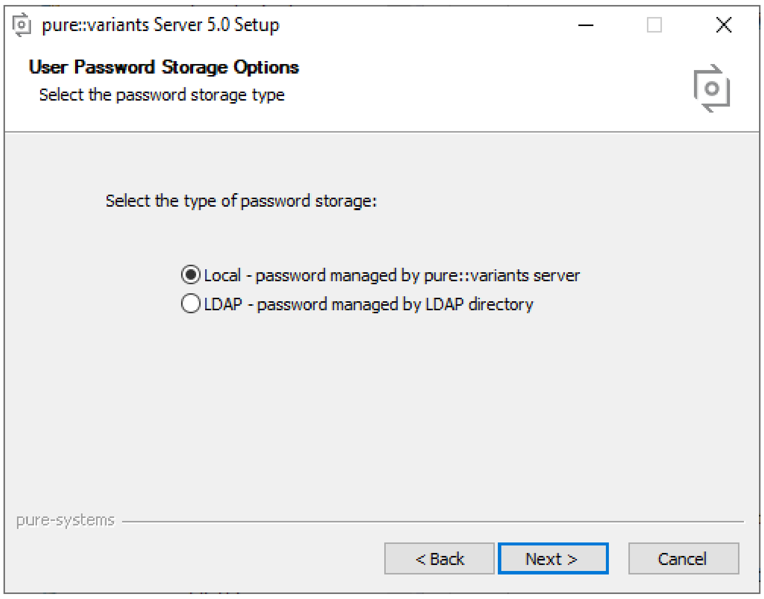
Protect the web browser access with a password.

Password:

Retype Password:

The model server provides a web interface. The web interface can be enabled or disabled on this page. Please secure the web interface with a password. The web interface should be used with connection encryption only, in order to prevent possible risk of password theft. Click *Next*.

Figure 66. Setup Model Server Password Management


pure::variants Server 5.0 Setup

User Password Storage Options

Select the password storage type

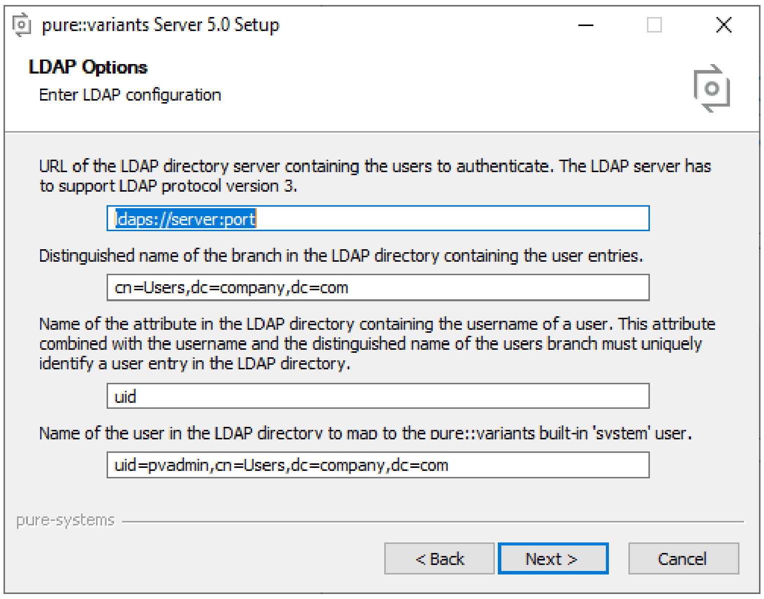
Select the type of password storage:

[✓ Local - password managed by pure::variants server]

LDAP - password managed by LDAP directory

The model server is capable of managing password locally, or can access a LDAP directory for user authentication. Please chose the password management method. Click *Next*.

Figure 67. Setup Model Server LDAP Settings


pure::variants Server 5.0 Setup

LDAP Options

Enter LDAP configuration

URL of the LDAP directory server containing the users to authenticate. The LDAP server has to support LDAP protocol version 3.

ldaps://server:port

Distinguished name of the branch in the LDAP directory containing the user entries.

cn=Users,dc=company,dc=com

Name of the attribute in the LDAP directory containing the username of a user. This attribute combined with the username and the distinguished name of the users branch must uniquely identify a user entry in the LDAP directory.

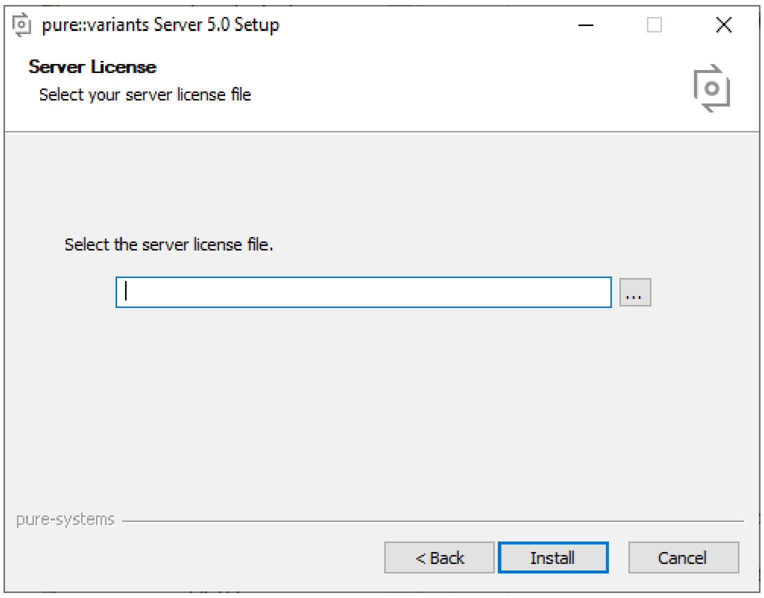
uid

Name of the user in the LDAP directory to map to the pure::variants built-in 'system' user.

uid=pvadmin,cn=Users,dc=company,dc=com

This page allows you to setup your specific LDAP settings. Click *Next*.

Figure 68. Setup Model Server License File


pure::variants Server 5.0 Setup

Server License

Select your server license file

Select the server license file.

As the last step, the license file for the pure::variants model server needs to be specified. Clicking *Install* will start the installation process.

If the Windows service option was chosen, the model server will start automatically after successful installation.

Install from Archive

If using the installer is not an option, the server is also distributed as a compressed archive file.

Download the pure::variants Database server archive from the pure::variants product web page. Go to http://www.pure-systems.com/pvde-update/.

The product download pages are protected by a password. You need to login by using the email address and the registration number from the server license file provided to you by pure-systems.

Create a directory for the pure::variants model server on your disk and extract the contents of the archive into that folder. For Linux systems a recommended location is the directory */opt/pure-variants*. On Windows a recommended location is the directory *C:\Program Files\pure-systems\pure-variants_Server_5.0*

Create a directory for the license files. A proper location for Linux systems is */opt/pure-variants/licenses* and on Windows *C:\Program Files\pure-systems\pure-variants_Server_5.0\licenses*. Copy the provided server license file into that folder.

Open the script *start.bat* on Windows resp. *start.sh* on Linux in a text editor. The script is located in the *server* sub-directory. Update the following variables to match your environment:

HOSTNAME: The hostname or IP address of the server machine the server should bind to.

PORT: The TCP port used for communication.

LICENSEDIR: The folder with the license files.

Clients must be able to connect to this port on the server machine. Therefore this port must not be blocked by a firewall or used by another service. A port number can range from 0 to 65535. Port 80 is the standard port for the HTTP and 443 for the HTTPS protocol. Using these standard ports can help with firewall problems.

The server can be now started by running script *start.bat* on Windows resp. *start.sh* on Linux. Please consult your system documentation how to add this script to the systems start sequence to ensure the server is started automatically after system reboot. You are now able to connect to the pure::variants Model Server with your pure::variants Enterprise clients.

On Windows the pure::variants server can be registered as a service. Selecting this option ensures that the pure::variants server always runs after reboot even when no user is logged in. To register the puer::variants server as a service, use the command line option */install*. If installing the pure::variants license server as a service, a service name has to be given with command line option */servicename*. A service description can be added with command line option */servicedesc*.

```
<license server install path>/server/variantsd.exe /install
/servicename "pure::variants License Server" /servicedesc "The license server is providing
licenses for the pure::variants clients"
```

Enable pure::variants Model Server Web Interface

The pure::variants model server includes an optional web interface for monitoring and interacting with the server.

To enable the server's web interface on Windows, add the following options to the server command at the end of the *server/start.bat* script:

```
/enableweb
/webpwd PASSWORD
```

To enable the server's web interface on Linux, add the following options to the server command at the end of the *server/start.sh* script:

```
--enableweb
--webpwd PASSWORD
```

The server's web interface should be always protected by a password. Please use a secure password for this.

5.1.2. Basic pure::variant Model Server Setup

The initial configuration of the pure::variants server only knows one account, i.e. the special account “system”. If the server is configured to use “local” authentication, which is the default, this account has a default password which is the same for all pure::variants server installations. Thus, it **MUST BE CHANGED** immediately after starting up the server for the first time. Keeping this password renders the server insecure. The password is “sXg9C58JcmPB“ (without the double quotes).

Start the pure::variants server as follows. If installed as service, start the service manually using the Windows management console, or reboot. If not installed as service, start the server from the Start menu (in this case the server terminates upon logout).

The process of changing the password of a user is documented in the online help of the pure::variants client (*Help -> Help Contents -> pure::variants Server Administration plug-in manual*). It requires the import of the “ADMIN” project from the server. See the online documentation for information on the import process.

The recommended next step is to setup an administrative user different from the special “system” user. Using the newly created server administrator account, new users and roles may be added.

5.2. Update pure::variants Model Server

5.2.1. Update with Windows Installer (Windows Only)

Updating an existing pure::variants model server works exactly the same as a clean installation of the pure::variants model server. Please see Section 4.1.1, “Install with Windows Installer (Windows only)”. Additional settings performed in the server configuration file will remain in the server configuration file after updating.

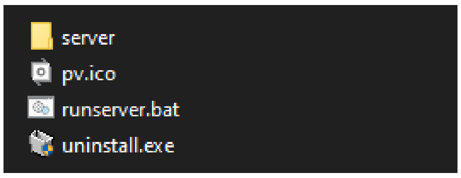
5.2.2. Update with Archive

Updating an existing pure::variants model server instance works exactly the same as a clean installation of the pure::variants model server. Please see Section 4.1.2, “Install from Archive”.

5.3. Uninstall pure::variants Model Server

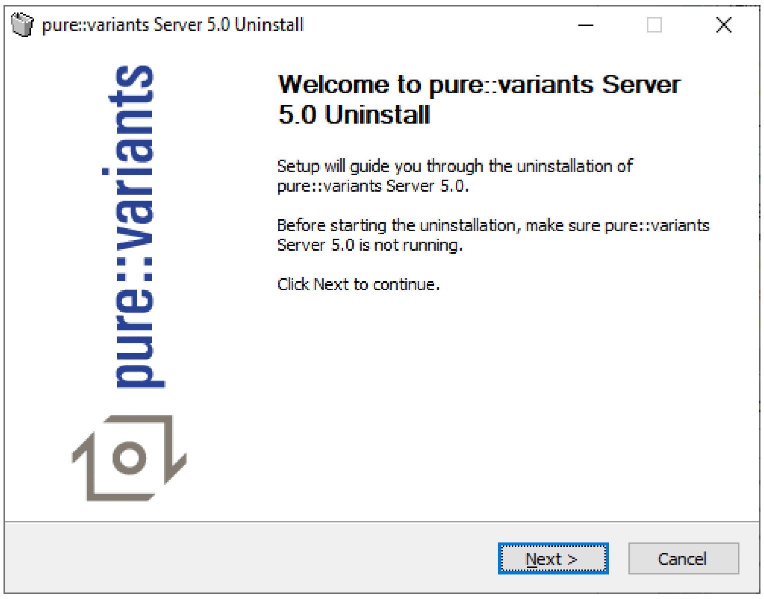
The uninstaller for the pure::variants model server can be started in two different ways. The first one is to go to the Windows *Add or remove programs* application and search for *pure::variants Server 5.0* and start the uninstaller by using the *Uninstall* action. The uninstaller requires Administrator privileges.

Figure 69. pure::variants Model Server Uninstaller



The second possibility is to navigate to the pure::variants model server installation folder and start the uninstaller by double-clicking it.

Figure 70. pure::variants Model Server Uninstaller


pure::variants Server 5.0 Uninstall

Welcome to pure::variants Server 5.0 Uninstall

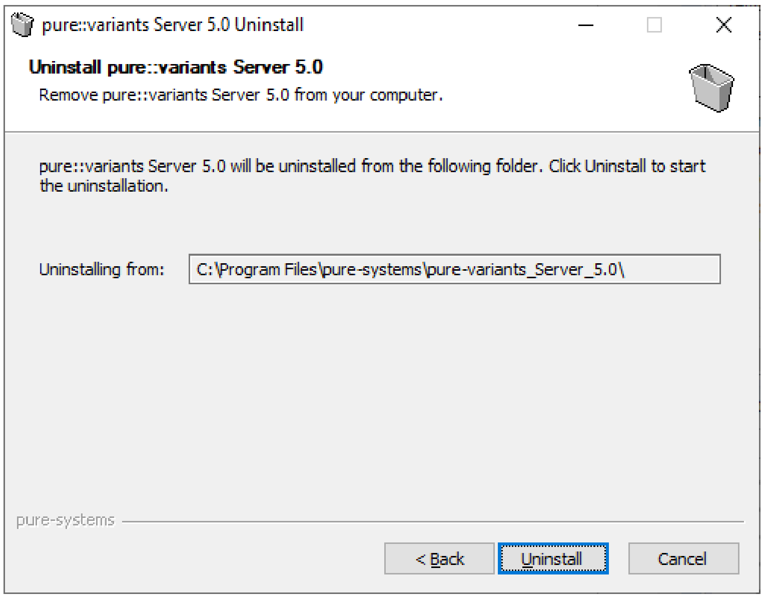
Setup will guide you through the uninstallation of pure::variants Server 5.0.

Before starting the uninstallation, make sure pure::variants Server 5.0 is not running.

Click Next to continue.

Click *Next*.

Figure 71. Uninstall from



Uninstall pure::variants Server 5.0

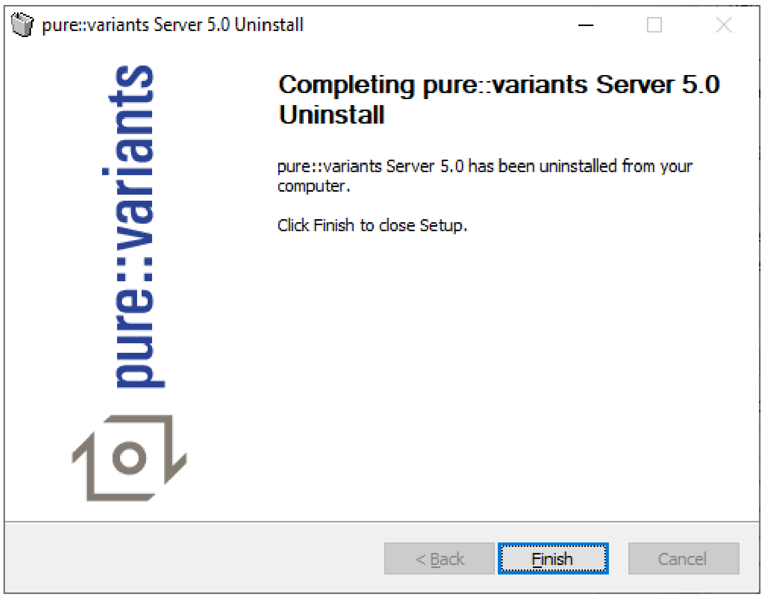
Remove pure::variants Server 5.0 from your computer.

pure::variants Server 5.0 will be uninstalled from the following folder. Click Uninstall to start the uninstallation.

Uninstalling from: C:\Program Files\pure-systems\pure-variants_Server_5.0\

Click *Uninstall* to start the uninstall process.

Figure 72. Completing Uninstall



Completing pure::variants Server 5.0 Uninstall

pure::variants Server 5.0 has been uninstalled from your computer.

Click Finish to close Setup.

The deinstallation is succesfully finished. Click *Finish* to close the uninstaller.

5.4. Location of the Server Configuration File

5.4.1. On Windows

You find the server configuration file “server.cfg” in the directory “pure-variants”, which either is located in the sub-directory “server\etc” of the installation directory of the pure::variants Server (e.g., “C:\Program Files\pure-systems\pure-variants_Server_5.0\server\etc”), or in the home directory of the current user.

5.4.2. On Linux

You find the configuration file “server.cfg” in the directory “pure-variants”, which either is located in the sub-directory “server/etc” of the installation directory of the pure::variants Server (e.g., “/opt/pure-systems/pure-variants_Server_5.0/server/etc”), or in directory “/etc”, or in the home directory of the current user.

5.5. Explanation of Installation Options and Parameters

All options are stored as command line settings in the server configuration file “server.cfg”.

5.5.1. Server Network Options

The pure::variants server uses the SOAP over HTTP/HTTPS protocol to communicate with its clients. The Server Network Options define the URL that users have to enter to reach a server installation. The URL for a server looks like this:

<Protocol>//<Address>:<Port>

Address: The hostname of the server machine. This will be the <Address> part of the URL.

Encryption of client/server communication: When the HTTPS protocol is used, all communication between clients and server is encrypted. This requires an X.509 certificate for the server. Certificate setup is done on a separate page in the installer.

Port: The TCP Port used for communication. Clients must be able to connect to this port on the server machine. Therefor this port must not be blocked by a firewall or used by another service. A port number can range from 0 to 65535. Port 80 is the standard port for the HTTP and 443 for the HTTPS protocol. Using these standard ports can help with firewall problems.

5.5.2. Install as Windows Service

The pure::variants server may be started either manually by the user from the Start menu, or automatically as Windows Service when Windows starts. Selecting this option applies the required changes to the Windows registry to register pure::variants as a service. This requires administrative rights during installation. Selecting this option ensures that the pure::variants server always runs after reboot even when no user is logged in.

5.5.3. HTTPS Server Options

The server includes the option to secure the network communication between the server and clients so that the transmitted data cannot be intercepted by others.

The encrypted communication makes use of an X.509 certificate containing the public and private parts of the certificate in one file. Smartcard-based certificates are currently not supported.

Such a certificate either can be obtained from a (official) certification authority (preferred) or can be a self-signed certificate. If the certificate is not issued by an official certification authority, all pure::variants users have to register this certificate or the root certificate of the internal certification authority in the pure::variants (Java) key store. Instructions how to do this are given below.

A self-signed certificate should be used only as temporary solution until an official certificate has been provided. It can be created for instance using the *openssl* tool like this:

```
openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -out certificate.pem
```

Preparation of the PEM file to be used with pure::variants Server

The PEM formatted file containing the public server certificate and private key should look like this.

```

            -----BEGIN ENCRYPTED PRIVATE KEY-----
```

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
...
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-----END ENCRYPTED PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
...
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-----END CERTIFICATE-----
```

If you have one file for the private key and one for the public certificate, then copy both together in one file using a text editor.

For security reasons, do not place this file in a path that is exposed to users other than the one running the pure::variants Server.

Configuration of the pure::variants Server for encrypted communication using a server configuration file

To enable encryption for an already installed pure::variants Server, the server configuration needs to be extended.

Open the configuration file in a text editor. If the file does not exist, create it. Add the following lines to the contents of the file.

```
/sslkeyfile "path\to\server-cert.pem"
/sslpassword "password of private key"
```

Replace the path to the PEM file and the password of the private key accordingly.

Configuration of the pure::variants Server for encrypted communication using command line options

To achieve this you have to modify the pure::variants server startup command line to include a reference to the key.

To enable the use of the certificate on Windows, add the following options to the server command at the end of the *start.bat* script:

```
/sslkeyfile <path_to certfile.pem>
/sslpassword certpassword
```

To enable the use of the certificate on Linux, add the following options to the server command at the end of the *start.sh* script:

```
--sslkeyfile <path_to certfile.pem>
--sslpassword certpassword
```

Usage of a self-signed server certificate with the pure::variants Client

The pure::variants Client uses the Java services to encrypt the communication with the pure::variants Server. Java requires all server certificates to be signed by a known certification authority, or to be registered as trusted certificate using a separate trust store.

If the pure::variants Server uses a self-signed certificate, all pure::variants Clients need to import this certificate into the pure::variants Client's trust store. The following steps show how to do this using the Java tools.

Step 1: Add the public server certificate to a Java Key Store

The tool keytool is part of the Java tools distributed with the JRE or JDK. It is located in the bin directory inside the Java installation.

Change to the home directory of the user running the server (%USERPROFILE%), and execute following command.

```
keytool -import -alias psroot -file certfile.pem
```

If you are asked to trust the certificate, then answer with Yes. This will create the file .keystore in the current directory. You may have to add further trusted certificates to this trust store if you need to access other remote sites using HTTPS from within pure::variants.

Step 2: Use the Java Key Store with the pure::variants Client

The two most common ways to provide the pure::variants Client with the server certificate are to extend the command line in the Desktop or Windows Start Menu link used to start the pure::variants Client, or to extend file *eclipse.ini* in the pure::variants Client installation directory.

To extend the command line, open the "Properties" dialog of the pure::variants Client link and append the following after the *-vmargs* option.

```
-Djavax.net.ssl.trustStore="%USERPROFILE%\.keystore"
-Djavax.net.ssl.trustStorePassword=keystorepassword
```

Replace the key store path and password accordingly.

Step 3: Use the Java Key Store with the pure::variants in-tool integration

If any in-tool integration (e.g., pure::variants integration for IBM Rational DOORS, or Microsoft Office) is used, the *trustStore* and *trustStorePassword* properties also need to be set for in-tool integrations. To do this, append

```
javax.net.ssl.trustStore=%USERPROFILE%/.keystore
javax.net.ssl.trustStorePassword=certpassword
```

to file:

```
C:\ProgramData\pure-variants-5\pv.properties
```

If "pv.properties" does not exist, create it and the necessary folders. Please note that slashes in the trustStore path need to be either forward slashes, or escaped slashes ("/" or "\\"). Otherwise the path cannot be read.

```
“C:\\” => “C\:\\” resp. “C:/” => “C\:/”
```

Please remember not to distribute the certificate file with the public and private part to the users! Only the public part of the certificate is required in this step! If an empty password was given in step 1 the second part of the command line can be omitted. The keystore file created in step one can be put on a shared directory and used by all users. In this case the path in the first part of the command line has to point to the shared location of the key store file.

5.6. Setup Authentication for pure::variants Model Server

5.6.1. Windows Authentication

The pure::variants Server can be configured to authenticate users using Windows Authentication. This authentication mechanism automatically chooses Kerberos or NTLM authentication depending on the configuration of the Windows host system. No password is required in order to access the server (single sign-on).

For this authentication mechanism to work, the user's client computer and the computer running the pure::variants Server need to be member of the same Windows domain, or at least need to have the same Windows accounts set up.

Windows Authentication is enabled for the pure::variants Server by setting command line option **/logon** to value **windows**.

In order to login to the pure::variants Server as the superuser “system”, an existing Windows user can be configured to act as user “system” using command line option **/systemuser**. If for instance user “john” has been decided to be the pure::variants Server’s superuser, then **/systemuser john** has to be added to the pure::variants Server’s configuration. If then user “john” successfully authenticates at the pure::variants Server, he automatically becomes the superuser “system”.

5.6.2. Open ID Connect Authentication

The pure::variants Server can be configured to authenticate users using an Open ID Connect provider. This enables single sign-on inside Eclipse between different applications using the same provider.

Open ID Connect authentication is enabled for the pure::variants Server by setting the command line option **/logon** to the value **openid**. Before the first start using Open ID Connect, the server needs to register at the provider. This is done by starting the server on the command line with the **/openidregister** option. The provider URL, the user and the password are set by the corresponding **/openidurl**, **/openiduser**, and **/openidpass** options. The following command line shows an example for a registration.

```
server/bin/variantsd.exe /config server/etc/pure-variants/server.cfg /openidregister
  /openidurl https://openid.company.com/oidc/endpoint /openiduser admin /openidpass password
```

If the connection to the Open ID Connect provider fails due to certificate or other SSL errors, you can use command line option **/openidignoresslerror** to let the pure::variants server ignore such errors. But beware, such errors may indicate that something is wrong with the Open ID Connect provider. Use this option with care. The same is true for the command line option **/openidnorevoke** which can be used to disable the check whether the certificate of the Open ID Connect provider has been revoked.

The registration data will be saved to file **openid.json** in the server configuration directory. After successful registration the server needs to be restarted.

If automatic registration at the Open ID Connect provider is not possible, the **openid.json** file has to be created by hand. This file has the following format:

```
{
    "client_id" : "insert here the client ID",
    "client_secret" : "insert here the client secret",
    "http_flags" : 0,
    "provider_uri" : "insert here the provider URL",
    "redirect_uris" :
    [
        "http://localhost/pv/openid"
    ],
    "scope" : "openid"
}
```

In order to get the client ID and client secret, you have to register the pure::variants server by hand at your Open ID Connect provider. Then copy the client ID and secret and insert both together with the URL of the Open ID Connect provider in the **openid.json** configuration file. If you configured a client URL other than "http://localhost/pv/openid", then you need to replace this URL in the configuration file too.

The **http_flags** setting in the configuration file can have a value between 0 and 3. Value 0 means that all security checks for the communication with the Open ID Connect provider are enabled. Value 1 means that certificate or other SSL errors are ignored, as if option **/openidignoresslerror** is used. Value 2 means that certificate revocation is not checked, as if option **/openidnorevoke** is used. Value 3 is the combination of values 1 and 2, as if both options **/openidignoresslerror** and **/openidnorevoke** are used.

Changes on the **openid.json** configuration file are not recognized by the pure::variants server when it is running. You need to restart the pure::variants server for the new configuration to apply.

The registration is stored until an unregistration is performed. For unregistration the **/openidunregister** option is used together with the **/openiduser** and **/openidpass** option.

5.6.3. LDAP Authentication

The pure::variants Server can be configured to authenticate users against an LDAP directory. Users known to the pure::variants Server can then login to the server using their LDAP password. The server will use the provided username and password to bind to the LDAP directory. If this succeeds, the user is considered to be authenticated.

For this to work the LDAP directory server needs to support version 3 of the LDAP protocol. It is tried at most 3 times to access the LDAP directory for each bind attempt. If the LDAP directory does not respond within 60 seconds, the bind attempt is aborted and the authentication of the user will not succeed.

To use LDAP authentication you either have to enable and configure it during installation of the pure::variants Server, or add corresponding options to the configuration file of the pure::variants Server. You find the configuration file either in the installation directory of the pure::variants Server, e.g. “C:\Program Files\pure-systems\pure-variants_Server_4.0\server\etc\pure-variants\server.cfg”, or in the home directory of the user, i.e. “%APPDATA%\pure-variants\server.cfg”.

Following options control the LDAP authentication performed by the pure::variants Server.

/logon ldap

The default authentication scheme of the pure::variants Server is “local”, which means that the passwords of the users are managed by the pure::variants Server itself in a password file or database. With the logon type “ldap” the pure::variants Server does not manage the passwords by itself but uses an LDAP directory server to authenticate users. Note that in this case it is not possible to change the password of users in pure::variants because the passwords are managed by the LDAP directory and need to be changed there.

/ldapurl [URL]

The URL of the LDAP directory server containing the users to authenticate. The format of this URL is defined by RFC 2255 and basically looks like this:

- ldap://server:port
- ldaps://server:port

Beside protocol “ldap” also protocol “ldaps” for secure LDAP connections is supported by the pure::variants Server. It is strongly recommended to use the “ldaps” protocol if supported by the LDAP directory server. The default port when using the “ldap” protocol is 389, and 636 when using the “ldaps” protocol.

/ldapusersdn [DN]

The sub-tree in the LDAP directory containing the user entries. It has to be given as full distinguished name (DN) according to RFC 4514 and basically looks like this:

- cn=Users,dc=company,dc=com

If no LDAP bind user is given (see option /ldapbinduser), it is expected that all user entries are listed flat in this sub-tree. If an LDAP bind user is given, then nested organization of user entries is supported. The pure::variants Server will then search the whole sub-tree to find a matching user for a given username.

/ldapuidattr [NAME]

The name of the LDAP attribute containing the username of a user in the LDAP directory. Example:

- uid

To authenticate a user against the LDAP directory, the full distinguished name of the user entry is needed. This distinguished name uniquely identifies the user in the LDAP directory and consists of the users sub-tree distinguished name (see option /ldapusersdn), the username, and this LDAP attribute (e.g. uid=username,cn=Users,dc=company,dc=com), containing further name parts if the users are organized in a nested instead of a flat tree structure (see option /ldapbinduser).

/ldapsysuser [DN]

A user from the LDAP directory to be used whenever a login as the pure::variants built-in “system” user is requested. As a consequence the password of this LDAP user needs to be used instead the documented “system” user's default password. If the user is not located in the users sub-tree of the LDAP directory (see option /ldapusersdn), then the full distinguished name of the user needs to be given. Otherwise the simple username can be used. Examples:

- uid=pvadmin,cn=Users,dc=company,dc=com
- pvadmin

/ldapbinduser [DN]

A user from the LDAP directory with search rights on the username attribute (see option /ldapuidattr) in the users sub-tree of the LDAP directory (see option /ldapusersdn). Example:

- uid=pvadmin,cn=Users,dc=company,dc=com

This option is not needed if the user entries are organized flat in the LDAP directory. But if they are organized in a nested tree structure, then this user is needed by the pure::variants Server to find users by their username.

/ldapbindpass [PASSWORD]

The password of the LDAP user configured with option /ldapbinduser.

Verify LDAP Configuration

It is recommended that a changed LDAP configuration is verified before the pure::variants Server is started. Otherwise it may not be possible for any user to login to the pure::variants Server.

For this purpose the pure::variants Server provides the ldaptest tool, located in the server\bin sub-directory of the pure::variants Server installation directory.

This tool supports the same command line options and configuration file settings as the pure::variants Server. If all settings have been done in the server configuration file, then the tool can be run without further arguments. Just double-click it or open a command prompt in the server\bin directory and enter:

ldaptest

Please follow the instructions on the screen. The tool tries to authenticate a given user that must exist in the LDAP directory. And it tries to authenticate the system user in the same way. If an LDAP bind user is given, then the tool tries to search a given user in the LDAP directory using the bind user before the authentication is tested.

This is an example of running the tool using command line arguments:

ldaptest /ldapurl "ldap://ldap.company.com:389" /ldapusersdn "cn=Users,dc=company,dc=com" /ldapuidattr "uid" /ldapsysuser "pvadmin"

The LDAP configuration is correct if the tool reports that the test successfully finished.

5.7. Server Command Line Options

Following list describes the options that can be added to the pure::variants server command line or configuration file.

See Section 5.4, “Location of the Server Configuration File” to find the configuration file.

Option	Description
-L, /loglevel [LEVEL]	Level for server logging (0-9).
-l, /logfile [FILE]	File for the server logging output.

-r, /rmlog	Remove the old log file on startup.
/config [FILE]	Path to the server configuration file listing command line options one by line.
-p, /port [PORT]	Port for the SOAP server connection (default 80).
-t, /clienttimeout [SECONDS]	Timeout in seconds before killing dead client connections (default 900 = 15min).
-a, /address [ADDRESS]	Address to bind for the SOAP server connection (default 0.0.0.0). Accepts IP address as well as host name.
-i, /info	Print server information (XML) on startup.
-h, /help	Show the command help text.
/printinfo	Print server information (XML) and exit.
/enableweb	Enable the servers HTTP interface.
/webpwd [PASSWORD]	Password for accessing the HTTP interface.
/licenselog	Enable Logging for License Server
/licenseuserlog	Enable Logging for License Server with full user data
/licenseuserlist [PATH]	Location of the user license access control list.
/sslkeyfile [PATH]	X.509 certificate for encrypted communication.
/sslpassword [PASSWORD]	Password for the X.509 certificate if needed.

There are some command line options, which are available in Windows only.

Option	Description
/service	Start as Windows service.
/install	Install as Windows service.
/remove	Remove Windows service.
/servicename	Name for service to install/remove
/servicedesc	Description for service to install/remove

5.7.1. Model Server Command Line Options

Following list describes the database model server specific options that can be added to the pure::variants license server command line or configuration file.

See Section 5.4, “Location of the Server Configuration File” to find the configuration file.

Option	Description
-P, /plugindir [PATH]	Semi-colon separated list of additional server plugin directories.
/prolog [PATH]	Location of the prolog interpreter executable.
-R, /plprog [PATH]	Location of the prolog resource database (prologrc).
-S, /xsltdir [PATH]	Location of the directory containing the server XSLT scripts.
/license	Location of the license file.
/logon [LOGON,LOGON,...]	Logon types to support (local, ldap, windows, openid).
/allowemptypassword	Allow users with an empty password to logon.
/systemuser [USERNAME]	User name of the user who is mapped to the internal system user
/aliasuser	User name of the user who is allowed to use aliases

/ldapurl [URL]	LDAP server URL (“ldap://server:port” or “ldaps://server:port”)
/ldapusersdn [DN]	LDAP users branch distinguished name (e.g. “cn=Users,dc=company,dc=com”)
/ldapuidattr [NAME]	LDAP username attribute of users (e.g. “uid”, or “cn”)
/ldapsysuser [DN]	LDAP user mapped to “system” user (e.g. “cn=pvadmin,cn=Users,dc=company,dc=com”)
/ldapbinduser [DN]	LDAP bind user
/ldapbindpass [PASSWORD]	LDAP bind user password
/openidregister	Register at Open ID provider
/openidunregister	Unregister from Open ID provider
/openidurl	Open ID provider URL (https://server:port/basepath)
/openiduser	Open ID user for registration
/openidpass	Open ID user password for registration
/openidignoresslerror	Ignore certificate validation errors for Open ID provider URL
/openidnorevoke	Disable Open ID provider certificate revocation check

Database Model Server Command Line Options

Following list describes the pure::variants database model server specific options that can be added to the server command line or configuration file.

See Section 5.4, “Location of the Server Configuration File” to find the configuration file.

Option	**Description**
/odbcdsn [NAME]	Name of the ODBC data source.
/odbcuid [USERNAME]	Name of the database user.
/odbcpwd [PASSWORD]	Password of the database user.
/odbctimeout [SECONDS]	Timeout for database operations (defaults to 2 minutes).

Filebased Model Server Command Line Options

Following list describes the pure::variants filebased model server specific options that can be added to the server command line or configuration file.

See Section 5.4, “Location of the Server Configuration File” to find the configuration file.

Option	**Description**
/projectsdata [PATH]	Location of the directory which contains the projects.

5.8. Proxy Configuration

The pure::variants model server can be placed behind a reverse proxy in the same way as the pure::variants license server. See Section 4.10, “Proxy Configuration” for details.

6. pure::variants Connectors

6.1. Installation of pure::variants Connectors

Installing a connector into an existing pure::variants installation works the exact same way like installing the pure::variants client into an exsiting Eclipse instance. You just have to make sure the depemding pure::variants connectors are already installed or they have to be installed together with the new connector. See the section called “Using update site”.

6.2. pure::variants Connector for Capella

To install the pure::variants Connector for Capella, open Capella or Capella Studio and select **Help->Install New Software...**. Enter the address of your pure::variants update site. From the list of available features, select "pure::variants - Connector for Capella", "pure::variants - Connector for EMF Feature Mapping" and the pure::variants feature (e.g., "pure::variants - Enterprise").

For installation in Capella, the standard Eclipse update site needs to be set up. Otherwise the installation will fail due to missing dependencies. In Capella 1.1.x the Eclipse update site is configured per default. In Capella 1.2.x, the Eclipse Neon update site still needs to be added. To do that, open **Window->Preferences->Install/Update->Available Software Sites** and add the update site.

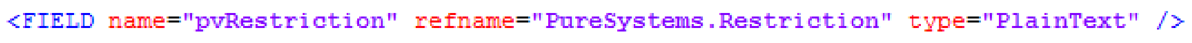
- For Capella 1.2.x use http://download.eclipse.org/releases/neon/
- For Capella 1.3.x use https://download.eclipse.org/releases/oxygen/
- For Capella 1.4.0 use https://download.eclipse.org/releases/2019-03/

6.3. pure::variants Connector for Team Foundation Server

For using the pure::variants Integration for Microsoft TFS the server has to be prepared and the pure::variants Integration has to be installed.

The work item types, which should be aware of variability information, must be configured with additional attributes. These attributes can be pvRestriction, pvConstraint, pvDefaultSelected and pvName. At least, the attribute pvRestriction should be created (as shown in Figure 73, “A XML configuration for pvRestriction field.”).

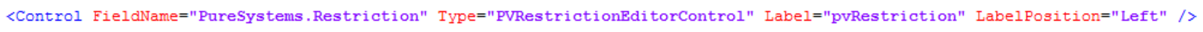
Figure 73. A XML configuration for pvRestriction field.



```
<FIELD name="pvRestriction" refname="PureSystems.Restriction" type="PlainText" />
```

Administrator: For having support while defining restrictions on work items, the control type for the pvRestriction attribute must be configured with "PVRestrictionEditorControl" (see Figure 74, “A XML configuration for pvRestriction field's control type.”).

Figure 74. A XML configuration for pvRestriction field's control type.



```
<Control FieldName="PureSystems.Restriction" Type="PVRestrictionEditorControl" Label="pvRestriction" LabelPosition="Left" />
```

6.4. pure::variants Connector for PTC Integrity

The pure::variants Connector for PTC Integrity as well as the pure::variants Integrity Integration expect a variant-related preparation of the solution item. For this several changes on the solution and settings are necessary, which are described in the following.

6.4.1. Add additional Fields for pure::variants

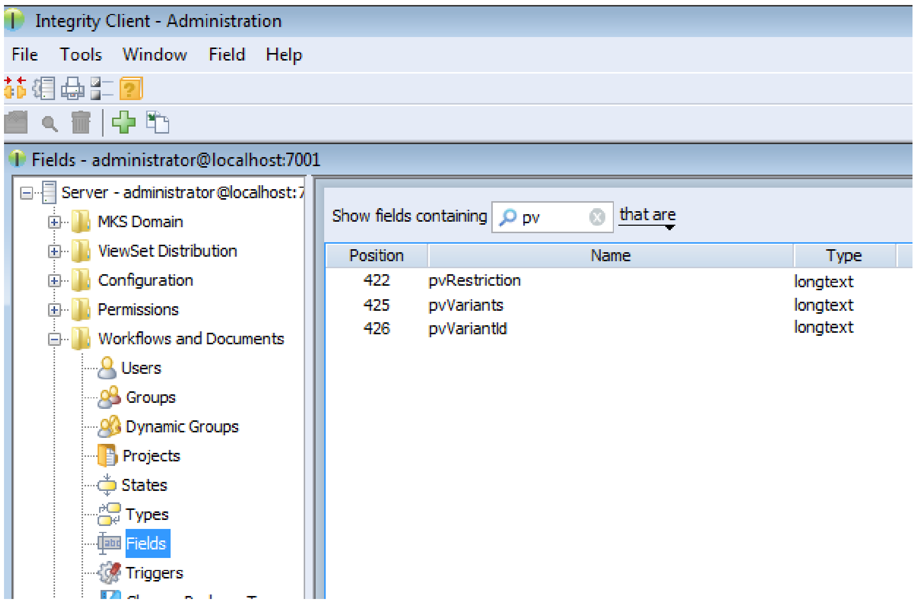
The following fields have to be created for the solution item, e.g. **MKS Solution**. In the PTC Integrity Administration open **Workflows and Documents** -> **Fields**. Add the fields by choosing **Create Field...** from the context menu.

Table 3. Additional fields

Field	Description
pvRestriction	This field is needed to store the restriction rule on a requirement. Set **longtext** as the type of the field, and ensure that the field is editable.

Field	Description
pvVariants	This field is needed to store the names of variants a requirement is part of. Set **longtext** as the type of the field. Ensure that the field is editable.
pvVariantId	This field is needed to store the hexadecimal encoded ID of the pure::variants variant description model which was used to create a requirement document variant in PTC Integrity. Set **longtext** as the type of the field. Ensure that the field is editable.

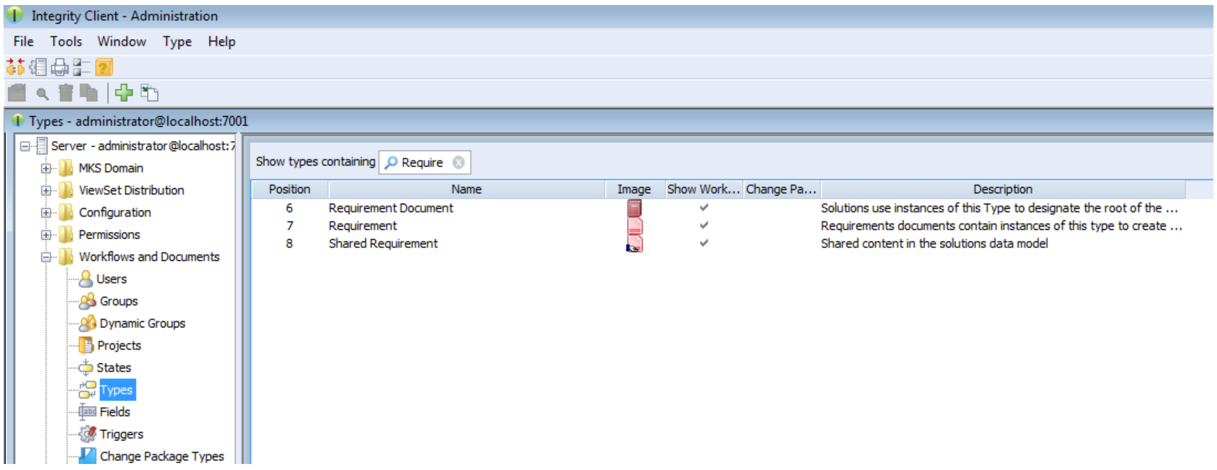
Figure 75. Added fields


Integrity Client - Administration
File Tools Window Field Help
Fields - administrator@localhost:7001
Server - administrator@localhost:7
MKS Domain
ViewSet Distribution
Configuration
Permissions
Workflows and Documents
Users
Groups
Dynamic Groups
Projects
States
Types
Fields
Triggers
Show fields containing pv that are

Position	Name	Type
422	pvRestriction	longtext
425	pvVariants	longtext
426	pvVariantId	longtext

Open **Workflows and Documents** -> **Types** in the PTC Integrity Administration and filter for all **Requirement** types.

Figure 76. Solution Types


Integrity Client - Administration
File Tools Window Type Help
Types - administrator@localhost:7001
Server - administrator@localhost:7
MKS Domain
ViewSet Distribution
Configuration
Permissions
Workflows and Documents
Users
Groups
Dynamic Groups
Projects
States
Types
Fields
Triggers
Change Package Types
Show types containing Require

Position	Name	Image	Show Work…	Change Pa…	Description
6	Requirement Document		✓		Solutions use instances of this Type to designate the root of the …
7	Requirement		✓		Requirements documents contain instances of this type to create …
8	Shared Requirement		✓		Shared content in the solutions data model

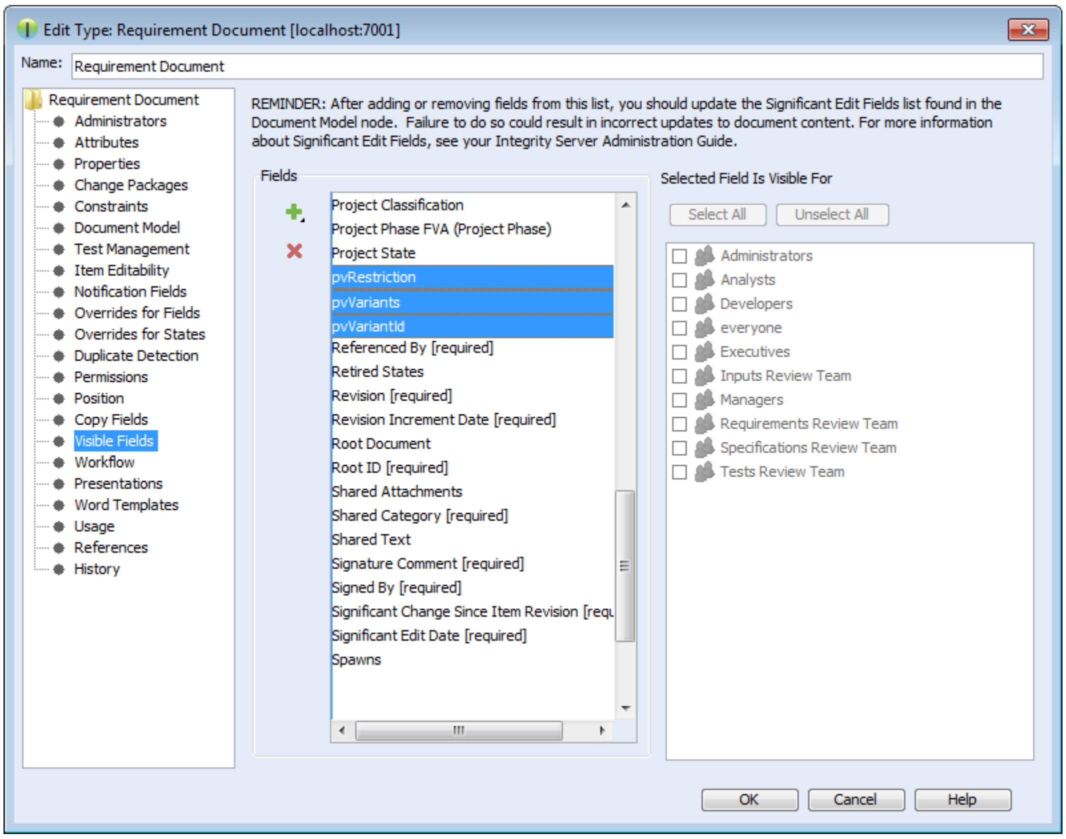
Add the new fields as **Visible Fields** to the following types.

Table 4. Types to add the fields for

Type	Fields to add
Requirements Document	pvRestriction, pvVariants, pvVariantId
Requirement	pvRestriction, pvVariants

Type	Fields to add
Shared Requirement	pvRestriction

Figure 77. Fields added to type Requirement Document



Edit Type: Requirement Document [localhost:7001]

Name: Requirement Document

REMINDER: After adding or removing fields from this list, you should update the Significant Edit Fields list found in the Document Model node. Failure to do so could result in incorrect updates to document content. For more information about Significant Edit Fields, see your Integrity Server Administration Guide.

6.4.2. Change Connector and In-Tool Integration Settings

pure::variants uses following settings in order to connect to PTC Integrity. This includes settings for the pure::variants Connector for PTC Integrity, which allows importing and exporting documents, as well as the settings for the In-tool Integration, which allows adding and changing restriction rules in PTC Integrity.

Table 5. Settings

Setting	Default Value	Description
Solution Item	MKS Solution	Used to get configuration properties
Document Type Field	Type	Used for the export of variant documents
Project Field	Project	Used for the export of variant documents
Document Fields	Document Short Title	Used to get several information from imported requirement documents, and to copy fields when exporting variant documents. The first field must always be the document title field
Document Title Field	Document Short Title	Used to get the document title from imported requirement documents, and to calculate the title of exported variant documents
Restriction Rule Field	pvRestriction	Used to get restriction rules from imported requirement documents, and to read and write restriction rules

Setting	Default Value	Description
Variant Enumeration Field	pvVariants	Used to store enumerated variant names in requirement documents
Variant Document ID Field	pvVariantId	Used to store the ID of pure::variants variant description models in variant requirement documents
Requirement Text Field	Text	Used while import to get the text of requirements from requirement documents

Some of these settings can be directly changed before importing and exporting requirement documents. Others can only be changed in the connector configuration file and in the solution type. The following table shows the property names used to change these settings in the configuration file and the solution type.

Table 6. Properties

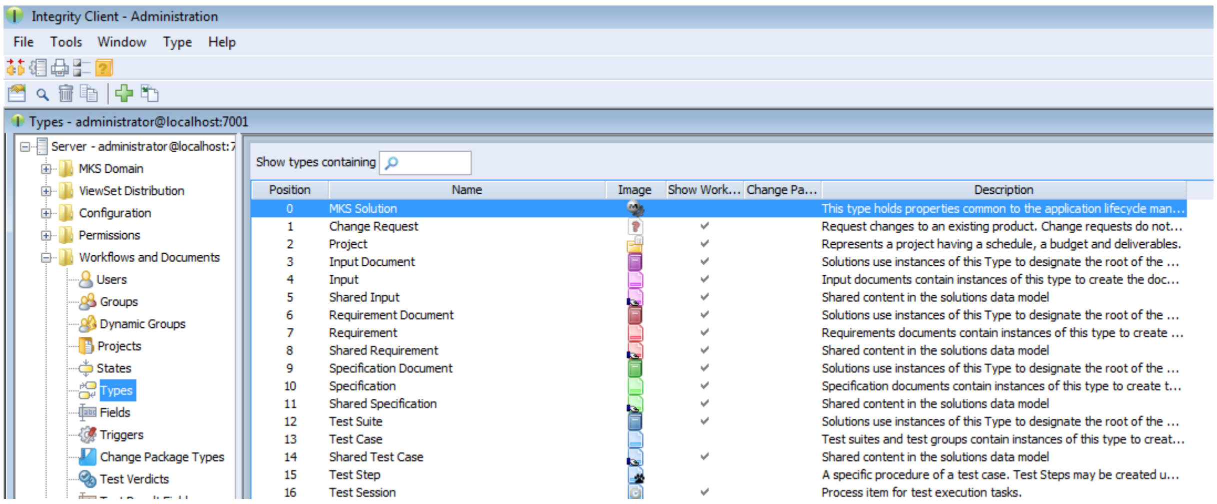
Setting	Config File Property	Solution Type Property
Solution Item	solutionType	
Document Type Field	fieldname.documenttype	PUREVARIANTS.TYPE.FIELD
Project Field	fieldname.documentproject	PUREVARIANTS.PROJECT.FIELD
Document Fields		PUREVARIANTS.VARIANT.FIELDS
Document Title Field	fieldname.documenttitle	PUREVARIANTS.TITLE.FIELD
Restriction Rule Field	attrname.restrictions	PUREVARIANTS.RESTRICTION.FIELD
Variant Enumeration Field	attrname.variants	PUREVARIANTS.VARIANTS.FIELD
Variant Document ID Field	fieldname.variantid	PUREVARIANTS.VARIANTID.FIELD
Requirement Text Field	fieldname.requirementtext	PUREVARIANTS.TEXT.FIELD

To change these settings in the connector configuration file, create the file **pvIntegrity.properties** in directory **%APPDATA%\pure-variants-5**. For each setting to change, add a line with the config file property of the setting assigned to the new value. To change for instance the default field used for storing restriction rules to **MyRestriction** and the default field used for storing the enumerated variants to **MyEnumeratedVariants**, you would add the following two lines to the configuration file (the comments are optional):

```
# Field used to store restriction rules
attrname.restrictions=MyRestriction
# Field used to store enumerated variant names
attrname.variants=MyEnumeratedVariants
```

To change these settings on the solution item, open the Administration of PTC Integrity. Switch to **Workflows and Documents** -> **Types** and select the solution type (e.g. **MKS Solution**).

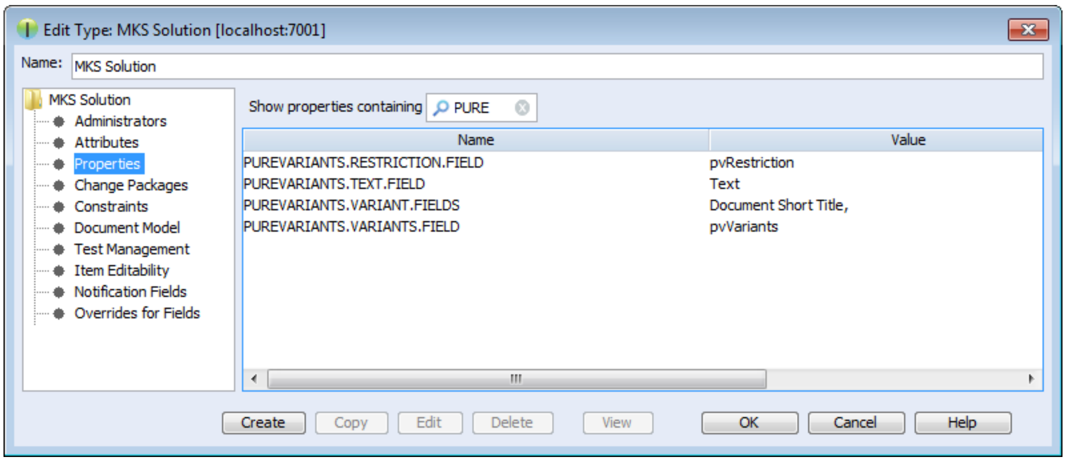
Figure 78. Type MKS Solution



Position	Name	Image	Show Work...	Change Pa...	Description
0	MKS Solution				This type holds properties common to the application lifecycle man...
1	Change Request		✓		Request changes to an existing product. Change requests do not...
2	Project		✓		Represents a project having a schedule, a budget and deliverables.
3	Input Document		✓		Solutions use instances of this Type to designate the root of the ...
4	Input		✓		Input documents contain instances of this type to create the doc...
5	Shared Input		✓		Shared content in the solutions data model
6	Requirement Document		✓		Solutions use instances of this Type to designate the root of the ...
7	Requirement		✓		Requirements documents contain instances of this type to create ...
8	Shared Requirement		✓		Shared content in the solutions data model
9	Specification Document		✓		Solutions use instances of this Type to designate the root of the ...
10	Specification		✓		Specification documents contain instances of this type to create t...
11	Shared Specification		✓		Shared content in the solutions data model
12	Test Suite		✓		Solutions use instances of this Type to designate the root of the ...
13	Test Case				Test suites and test groups contain instances of this type to creat...
14	Shared Test Case		✓		Shared content in the solutions data model
15	Test Step				A specific procedure of a test case. Test Steps may be created u...
16	Test Session		✓		Process item for test execution tasks.

Right-click the solution type and choose **Edit Type** from the context menu. Then switch to **Properties**, click the **Create** button and enter the solution type property name of the setting you want to change as name. Add the new default value as value and click **OK**.

Figure 79. Added Properties



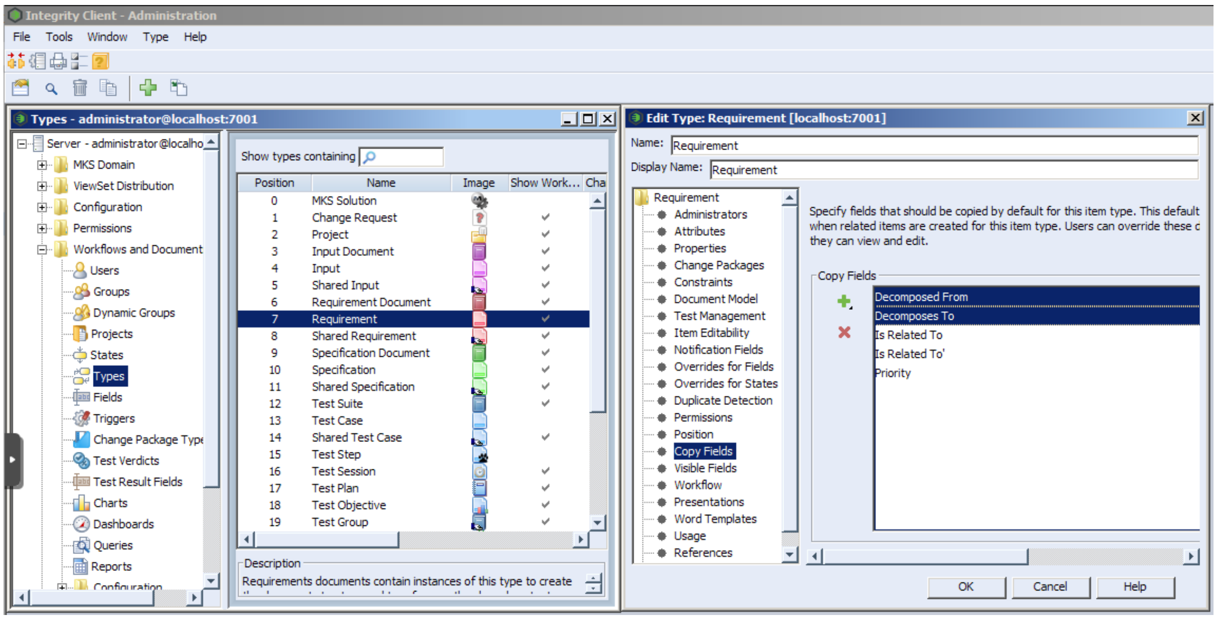
Name	Value
PUREVARIANTS.RESTRICTION.FIELD	pvRestriction
PUREVARIANTS.TEXT.FIELD	Text
PUREVARIANTS.VARIANT.FIELDS	Document Short Title,
PUREVARIANTS.VARIANTS.FIELD	pvVariants

6.4.3. Change Fields Copied for Variant Creation

Several fields of the original document are copied while creating a document variant. This includes the fields mentioned in section Section 6.4.1, “Add additional Fields for pure::variants” but also some fields that are copied by default. The list of fields that are to be copied by default can be configured by the Integrity administrator.

To additionally copy for instance decompose relationships, the administrator has to open the Administration of PTC Integrity. Then switch to **Workflows and Documents** -> **Types** and edit type **Requirement**. Open the **Copy Fields** list and add the fields **Decomposes To** and **Decomposed From**. This way fields could be added for the types **Requirements Document**, **Requirement**, and **Shared Requirement**.

Figure 80. Decompose Fields added



Types - administrator@localhost:7001

Show types containing

Position	Name	Image	Show Work...
0	MKS Solution		
1	Change Request		✓
2	Project		✓
3	Input Document		✓
4	Input		✓
5	Shared Input		✓
6	Requirement Document		✓
7	Requirement		✓
8	Shared Requirement		✓
9	Specification Document		✓
10	Specification		✓
11	Shared Specification		✓
12	Test Suite		✓
13	Test Case		
14	Shared Test Case		✓
15	Test Step		
16	Test Session		✓
17	Test Plan		✓
18	Test Objective		✓
19	Test Group		✓

Description: Requirements documents contain instances of this type to create

Edit Type: Requirement [localhost:7001]

Name: Requirement

Display Name: Requirement

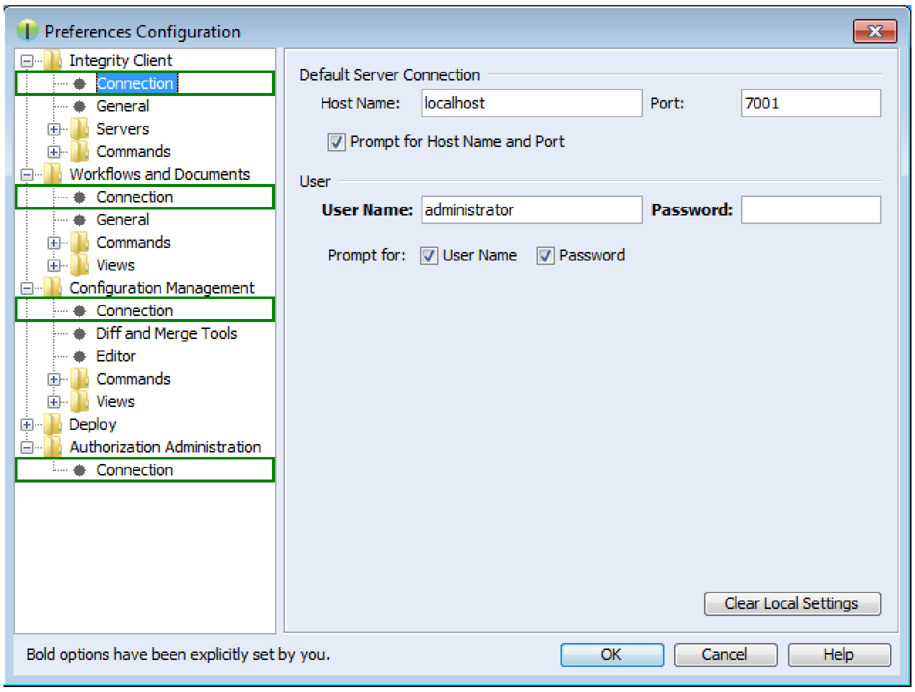
Specify fields that should be copied by default for this item type. This default when related items are created for this item type. Users can override these c they can view and edit.

Copy Fields: Decomposed From, Decomposes To, Is Related To, Is Related To', Priority

6.4.4. Enable PTC Integrity Client Access

The connector and the integration for PTC Integrity require access to the Integrity client in order to work. Start the PTC Integrity client and open menu **File** -> **Preferences**. For the entries **Integrity Client**, **Workflow and Documents**, **Configuration Management**, and **Authorization Administration** click the **Connection** section and enable **Prompt for Host Name and Port**, prompt for **User Name**, and prompt for **Password**.

Figure 81. pure::variants Credentials



Preferences Configuration

Default Server Connection

Host Name: localhost   Port: 7001

[✓] Prompt for Host Name and Port

User

User Name: administrator   **Password:**

Prompt for: [✓] User Name [✓] Password

Bold options have been explicitly set by you.

Note

These connection settings always are used even if you have the option "Prompt for Host Name and Port" enabled and change the connection settings in the corresponding dialog when importing documents from Integrity into pure::variants or exporting variants to Integrity.

6.5. Connector for IBM Rational Rhapsody

If you are working with Rhapsody Model Manager (RMM) projects, additional setup steps are needed.

First, you need to make sure that all required software is installed. That includes RMM (Architecture Management) on the Jazz server and Rational Team Concert (RTC) on your client machine. Also in Rhapsody, the Rhapsody Model Manager add-on needs to be installed.

6.5.1. Preparing IBM Rational Team Concert

In RTC two integrations need to be installed: The *IBM Rational Rhapsody integration for Rational Team Concert* and the *pure::variants Integration for RTC transformation*. The IBM Rational Rhapsody integration for RTC is needed for RTC to work with RMM projects. Please consult the RTC documentation for installation instructions.

The *pure::variants Integration for RTC transformation* is needed during transformation of RMM projects. Without it, the transformation will fail. To install the pure::variants Integration for RTC transformation, please open RTC and use **Help** > **Install New Software** to install all contents of archived update site `com.ps.consul.eclipse.rtc.integration.feature_[version].zip`. You can find the archived update site zip in your Rhapsody integration installation folder (default is `C:\Program Files\pure-systems\pv_Enterprise_5.0\com.ps.consul.eclipse.ui.rhapsody.integration`).

6.5.2. Preparing pure::variants

For the transformation of RMM projects to work, you still need to set the RTC executable location. You can do this in the pure::variants preferences at **Window** > **Preferences** > **Variant Management** > **Connector Preferences** > **Connector for IBM Rational Rhapsody**. Alternatively, you can define an environment or Java system variable that is named **PV_RTC_EXEC_PATH** and whose value points to the RTC executable location.

When using Rhapsody 9.0 or above, it is possible to run a transformation of RMM projects in offline mode. This means that the transformatin is carried out without starting RTC/EWM client. At the above mentioned preference page, you can select the checkbox to enable this feature. Alternatively, you can set an environment or Java system variable that is named **PV_RHAPSODY_RMM_OFFLINE_MODE** to true to set the offline mode.

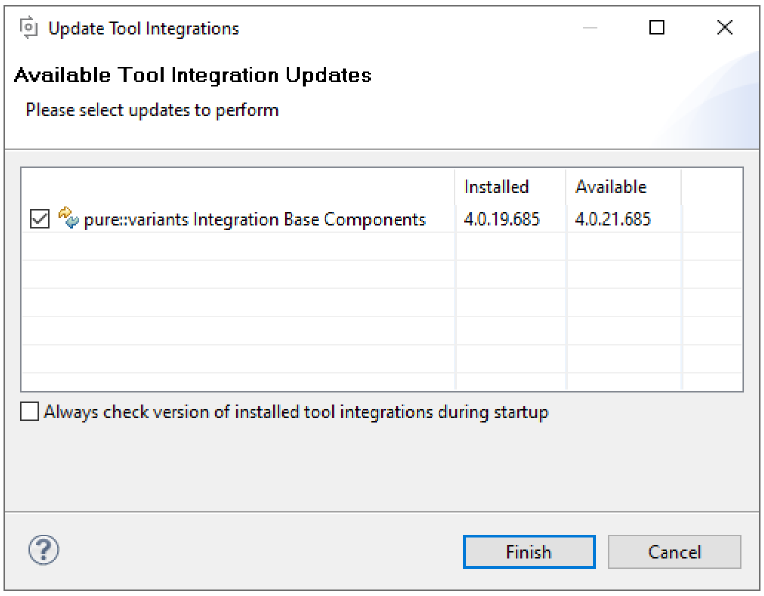
7. pure::variants Tool Integrations

7.1. Install pure::variants Tool Integrations

For using the pure::variants Integrations for several tools the integrations have to be installed. If pure::variants is installed using the Installer, the tool integrations are installed along with the corresponding pure::variants connector. If the installer was not used or the integration was not installed along with the corresponding connector then follow the next steps. Usually a dialog comes up after pure::variants starts informing about not installed or not up-to-date integrations.

If this dialog does not come up automatically it can be opened using the follwing menu entry in the pure::variants **Help** menu. Go to **Help**->**pure::variants** the item **Tool Integration Updates**.

Figure 82. Install Tool Integrations



Update Tool Integrations

Available Tool Integration Updates

Please select updates to perform

	Installed	Available
☑ pure::variants Integration Base Components	4.0.19.685	4.0.21.685

☐ Always check version of installed tool integrations during startup

A dialog comes up listing all available Tool Integrations. Just select the integrations you want to install or update and finish the dialog. pure::variants will guide you through the installation process. If an automatic update is possible, pure::variants will just perform the update, without showing an installer.

The option **Always check version of installed tool integrations during startup** enables a check if all available tool integrations are already installed and up to date. This check is performed each time pure::variants starts.

7.1.1. pure::variants Desktop Hub

If the pure::variants client was installed with the installer the pure::variants Desktop Hub is already installed along with the pure::variants client. If the installer was not used or the pure::variants Desktop Hub is missing for another reason the installation can be triggered from within the pure::variants client.

Please see Section 7.1, “Install pure::variants Tool Integrations” for installing the **pure::variants Integration Base Components**.

7.1.2. pure::variants Integration for Doors

Please see Section 7.1, “Install pure::variants Tool Integrations” for installing the integration executable.

After finishing the installation successfully, the Doors client has to be started with a command line option, which enables the Integration menu within Doors.

The command line should look similar to this: `doors -a "<Menu installation path>\pure-variants"`. Without this command line option the Integration cannot be triggered and so not be used.

On Windows platforms it is also possible to add the directory to the registry key `HKLM\Software\Telelogic\Doors\Doors version\Addins`.

1. Open the Registry editor
2. Browse to Doors installation in HKEY_LOCAL_MACHINE\SOFTWARE\Telelogic\DOORS\<DOORS version number>\Config
3. Right click config Key to add a new string value
    - Value Name set to **Addins**
    - Value Data set to the path of the pure::variants menu directory

With administrative access to the Doors installation the Add-In can also be installed for all users of this installation using the shared DXL library. See the Doors Help topic "Configuring Doors" for more information.

Note

This requires adaptions of the pure::variants Integration menu DXL scripts.

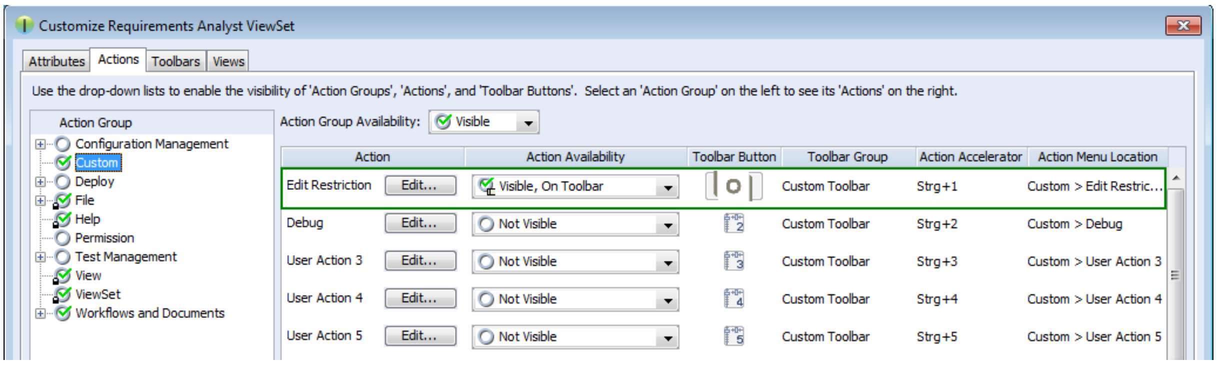
Now the Integration should be available in Doors. To verify if the installation was successful, open a Doors module and select from the menu **pure::variants** the item **Open pure::variants Integration**. If the pure::variants Integration window opens, the Integration was installed correctly.

7.1.3. pure::variants Integration for PTC Integrity

Please see Section 7.1, “Install pure::variants Tool Integrations” for installing the integration executable.

For starting the integration within PTC Integrity some additional steps have to be performed. Start the PTC Integrity client application. Open menu **ViewSet** -> **Customize** and switch to page **Actions**. Click on action group **Custom** to view the custom user actions.

Figure 83. Custom Actions



Customize Requirements Analyst ViewSet

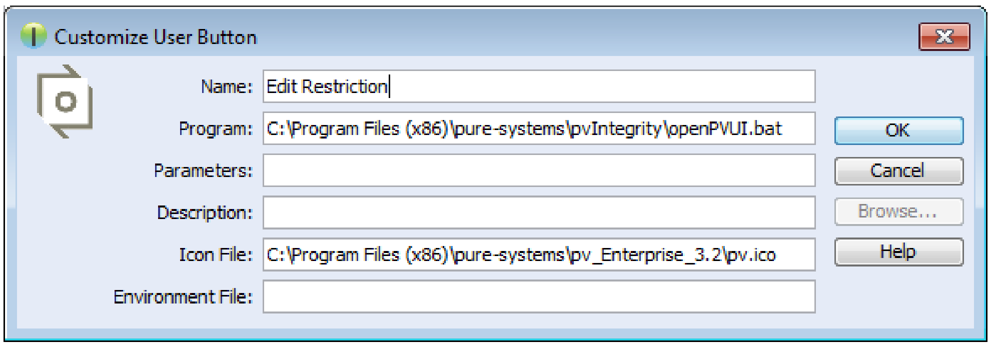
Use the drop-down lists to enable the visibility of 'Action Groups', 'Actions', and 'Toolbar Buttons'. Select an 'Action Group' on the left to see its 'Actions' on the right.

Action Group Availability: Visible

Action		Action Availability	Toolbar Button	Toolbar Group	Action Accelerator	Action Menu Location
Edit Restriction	Edit...	Visible, On Toolbar		Custom Toolbar	Strg+1	Custom > Edit Restric...
Debug	Edit...	Not Visible		Custom Toolbar	Strg+2	Custom > Debug
User Action 3	Edit...	Not Visible		Custom Toolbar	Strg+3	Custom > User Action 3
User Action 4	Edit...	Not Visible		Custom Toolbar	Strg+4	Custom > User Action 4
User Action 5	Edit...	Not Visible		Custom Toolbar	Strg+5	Custom > User Action 5

Click the button **Edit** of a user action to customize it. Name the action **Edit Restriction**. As the program to execute enter or browse to the path of file **openPVUI.bat** which is located in the installation path of the pure::variants Integration for PTC Integrity. There is also an icon file **pv.ico** in this directory you can use.

Figure 84. Custom Button "Edit Restriction"



Customize User Button

Name: Edit Restriction
Program: C:\Program Files (x86)\pure-systems\pvIntegrity\openPVUI.bat
Parameters:
Description:
Icon File: C:\Program Files (x86)\pure-systems\pv_Enterprise_3.2\pv.ico
Environment File:

7.1.4. pure::variants Integration for IBM Rational Rhapsody

Please see Section 7.1, “Install pure::variants Tool Integrations” for installing the integration executable.

To use the Integration, it still needs to be added to your Rhapsody project:

1. Open a Rhapsody project
2. Select **File** > **Add Profile to Model...**
3. Select the file "pvRhapsody.sbs" from the Integration installation directory

Now the Integration should be available for the given project. You can open the Integration window at **Tools** > **pure::variants**.

Per default the Integration is loaded when opening the Rhapsody project. If you want to only load the Integration when clicking **Tools** > **pure::variants**, you can edit file `pvRhapsody.prp` in the pure::variants Integration for Rhapsody installation folder. Open the file with a text editor and set property `showonstart` to `False`. After restarting Rhapsody, the Integration window should only open after clicking **Tools** > **pure::variants**.

7.1.5. pure::variants Integration for Enterprise Architect

Please see Section 7.1, "Install pure::variants Tool Integrations" for installing the integration executable.

Now the Integration should be available in Enterprise Architect. Select **Extensions**->**Add-In Windows** to open the Add-In window, which shows the pure::variants Integration user interface. Furthermore, you can enable or disable the Integration at **Extensions** -> **Manage Add-Ins...** (Since Enterprise Architect 14, you can find both entries in tab **Specialize**).

7.1.6. pure::variants Integration for Microsoft Office

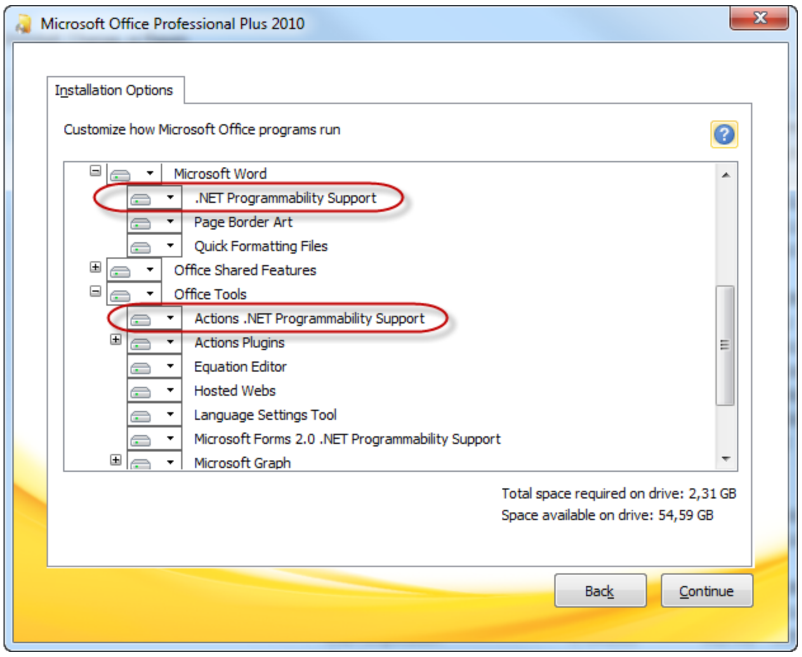
The Integration will not work if the following features are not installed with Microsoft Office:

- Microsoft Word / .NET Programmability Support (if using Microsoft Office Word Integration)
- Microsoft Excel / .NET Programmability Support (if using Microsoft Office Excel Integration)
- Office Tools / Actions .NET Programmability Support

They can be added to the Microsoft Office installation as follows:

1. Open the Windows Control Panel and navigate to **Programs and Features**
2. Right-click on your Microsoft Office Installation and select **Change**
3. Select **Add or Remove Features**
4. Add the features marked in Figure 85, "Adding Missing Features to the Office Installation" to your Office Installation.
5. Press **Continue** and close the Dialog.

Figure 85. Adding Missing Features to the Office Installation

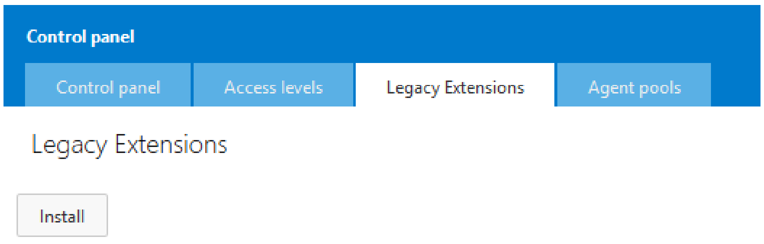

Microsoft Office Professional Plus 2010
Installation Options
Customize how Microsoft Office programs run
Microsoft Word
.NET Programmability Support
Page Border Art
Quick Formatting Files
Office Shared Features
Office Tools
Actions .NET Programmability Support
Actions Plugins
Equation Editor
Hosted Webs
Language Settings Tool
Microsoft Forms 2.0 .NET Programmability Support
Microsoft Graph
Total space required on drive: 2,31 GB
Space available on drive: 54,59 GB
Back
Continue

Now the installation of the pure::variants Integration for Microsoft Office should run successfully.

7.1.7. pure::variants Integration for Team Foundation Server

Administrator: At first, please download the pure::variants windows installer package from your pure::variants update site and extract the pure::variants TFS integration zip archive with name *com.ps.consul.web.ui.tfs2015-x.x.x.zip*. To install the integration navigate to the **Control panel**->**Legacy Extensions** and press **Install**. Thereby browse to your local copy of pure::variants TFS Integration zip archive.

Figure 86. Install pure::variants TFS Integration



Please ensure that the pure::variants TFS Integration is enabled, after installation.

7.1.8. Advanced Integration Setup

If you are using a pure::variants model or license server, establishing a connection with that server may need extra configuration. This may be the case, for example, if the server is located behind a proxy server or the communication with the server is encrypted and a self-signed certificate is used.

The steps needed to do in these cases differ, depending on which type of integration you are using. For java-based integrations, such as the Integration for IBM Rational Rhapsody and the Integration for PTC Integrity, it is necessary to specify proxy and certificate settings manually in pv.properties files. For .NET-based integrations (all other integrations), the Windows proxy and certificate settings are used automatically, so no additional setup is necessary.

Advanced Setup of Java-based Integrations

The following integrations are Java-based:

- pure::variants Integration for IBM Rational Rhapsody
- pure::variants Integration for PTC Integrity

All extra configurations in java-based integrations can be done by manually editing file `pv.properties`. You can find it in two different locations:

- If you want to configure the settings for all users on the machine, please edit

  ```
  %PROGRAMDATA%/pure-variants-5/pv.properties
  ```

- If you want to configure the settings only for the current user, please edit

  ```
  %APPDATA%/pure-variants-5/pv.properties
  ```

Note that:

- In case the file does not exist, you need to create it and any necessary folders first.
- Before editing `pv.properties`, make sure that no pure::variants Integration is running (e.g. Integration for Word, Excel, Doors, or the p::v Desktop Hub), otherwise your changes may be overwritten when closing the integration.

- Path delimiters in any paths you enter must be forward slashes or escaped backward slashes (`/` or `\\`). Otherwise the path cannot be read.
- All property names are case-sensitive.

Proxy Settings

The following properties can be set to configure your proxy settings.

Table 7. Proxy Settings

Property Name	Comments based on Java system property documentation
http.proxyHost	The hostname, or address, of the proxy server
http.proxyPort	The port number of the proxy server
https.proxyHost	The hostname, or address, of the proxy server in case HTTPS is used
https.proxyPort	The port number of the proxy server in case HTTPS is used
http.nonProxyHosts	Indicates the hosts that should be accessed without going through the proxy. Typically this defines internal hosts. The value of this property is a list of hosts, separated by the '|' character. In addition the wildcard character '*' can be used for pattern matching. For example http.nonProxyHosts=*.foo.com|localhost will indicate that every hosts in the foo.com domain and the localhost should be accessed directly even if a proxy server is specified. The default value excludes all common variations of the loopback address.
java.net.useSystemProxies	Set this to "true" to use Windows' global proxy settings (default: false), which are set in the Internet Explorer or in the Windows system settings. If one of the above properties is set, it overrides the respective Windows system property.

For example to use Windows' proxy settings, you would need to append this line to `pv.properties`:

```
java.net.useSystemProxies=true
```

Or to set all properties manually, you would need to append something like this:

```
http.proxyHost=YourHTTPProxyHost
http.proxyPort=80
https.proxyHost=YourHTTPSProxyHost
https.proxyPort=443
http.nonProxyHosts=*.foo.com|localhost
```

HTTPS Connection with License Server

The following HTTPS-related properties can be set. For more details, please refer to the respective Java system property documentation.

Table 8. HTTPS Settings

Property Name	Comments
javax.net.ssl.trustStore	Path to your trust store
javax.net.ssl.trustStorePassword	Password of your trust store
javax.net.ssl.trustStoreType	Trust store type (e.g. JKS)
javax.net.ssl.keyStore	Path to your key store
javax.net.ssl.keyStorePassword	Password of your key store
javax.net.ssl.keyStoreType	Key store type (e.g. JKS)
javax.net.debug	Activation of debug mode (e.g. "all" to write all possible debug logs)

Property Name	Comments
com.sun.net.ssl.checkRevocation	Enable certificate revocation checking

For example when using a self-signed certificate that is stored in trust store `D:/sandbox/servercert/cert-trusted.jks` you could append the following lines:

```
javax.net.ssl.trustStore=D\:/sandbox/servercert/cert-trusted.jks
javax.net.ssl.trustStorePassword=password
```

However, it is also possible to permanently accept a self-signed certificate when trying to first connect to your model or license server. pure::variants or the integrations will open an certificate acceptance dialog on the first connection attempt.

Advanced Setup of .NET-based Integrations

The following integrations are .NET-based:

- pure::variants Desktop Hub
- pure::variants Integration for Doors
- pure::variants Integration for Microsoft Excel and Microsoft Word
- pure::variants Integration for Enterprise Architect
- pure::variants Integration for Zuken CR-8000

Since pure::variants 4.0.19, the way the connection to pure::variants model or license servers is done has changed. Therefore, no advanced setup as for java integrations is necessary anymore. The proxy and certificate settings configured in Windows are used for the connection. So if the connection works in a browser that uses the Windows certificate and proxy settings (e.g. Chrome or Edge), the connection should work in all .NET-based integrations, too. The only exception from that rule are the supported security protocols:

Per default, the following security protocols are supported in .NET-based integrations: SSL3, TLS 1.0 - 1.3. To instead let Windows decide which protocols to support, you need to add registry entry "SystemDefaultTlsVersions" as documented here: https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client#configure-for-strong-cryptography.

If for some reason you want to switch back to the old server connection behavior as used in all java integrations, you can do that as described in the following section.

Switching Back to Previous Server Connection Behaviour

There are two ways to switch back to the previous server connection behaviour. Either you add the Windows environment variable `PV_FORCE_JAVA_SOAP_SERVER_CONNECTION` with value `true`, or you add line `forceJavaSoapConnection=true` to file `pv.properties` (see the section called “Advanced Setup of Java-based Integrations” for instructions how to edit `pv.properties`).

After switching back to the previous server connection behaviour, you may need to configure connection settings (e.g., proxy settings) in the same way as for java integrations.

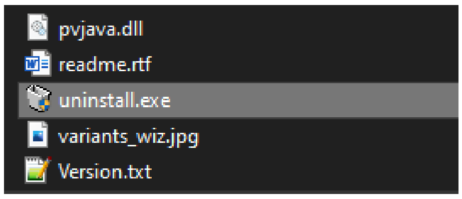
7.2. Update pure::variants Tool Integrations

To update an pure::variants tool integration the same mechanism as for installing a pure::variants tool integration is used. Please consult section Section 7.1, “Install pure::variants Tool Integrations” for a detailed description.

7.3. Uninstall pure::variants Tool Integrations

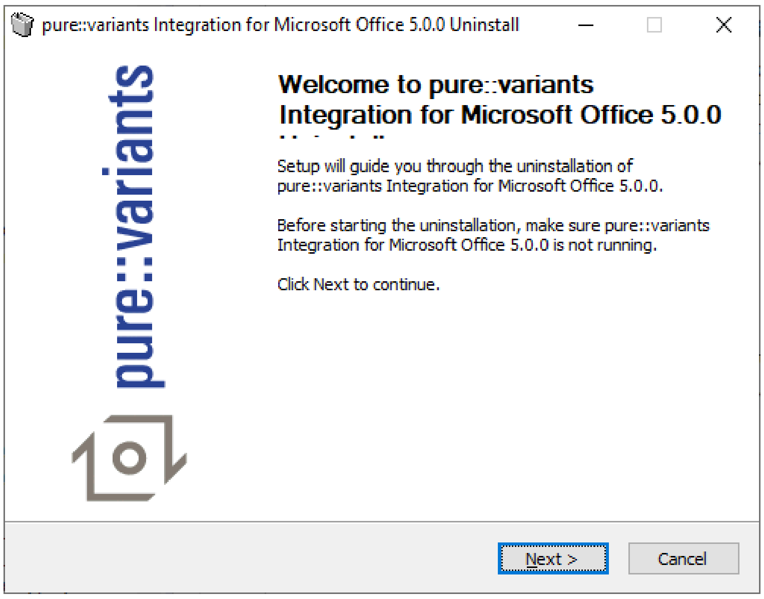
The uninstaller for the pure::variants integration can be started in two different ways. The first one is to go to the Windows *Add or remove programs* application and search for the pure::variants integration and start the uninstaller by using the *Uninstall* action. The uninstaller requires Administrator privileges.

Figure 87. pure::variants Integration Uninstaller


pvjava.dll
readme.rtf
uninstall.exe
variants_wiz.jpg
Version.txt

The second possibility is to navigate to the pure::variants client installation folder and start the uninstaller by double clicking it.

Figure 88. pure::variants Integration Uninstaller


pure::variants Integration for Microsoft Office 5.0.0 Uninstall

Welcome to pure::variants Integration for Microsoft Office 5.0.0

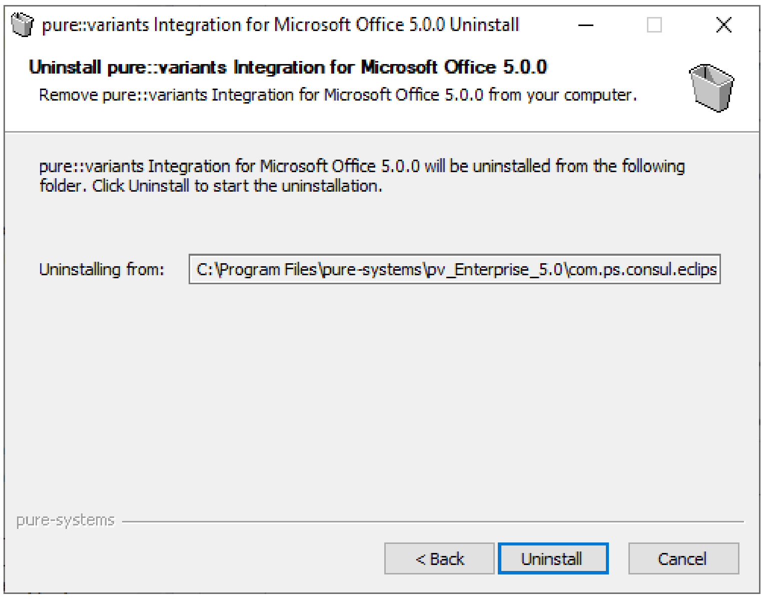
Setup will guide you through the uninstallation of pure::variants Integration for Microsoft Office 5.0.0.

Before starting the uninstallation, make sure pure::variants Integration for Microsoft Office 5.0.0 is not running.

Click Next to continue.

Click *Next*.

Figure 89. Uninstall from


pure::variants Integration for Microsoft Office 5.0.0 Uninstall

Uninstall pure::variants Integration for Microsoft Office 5.0.0

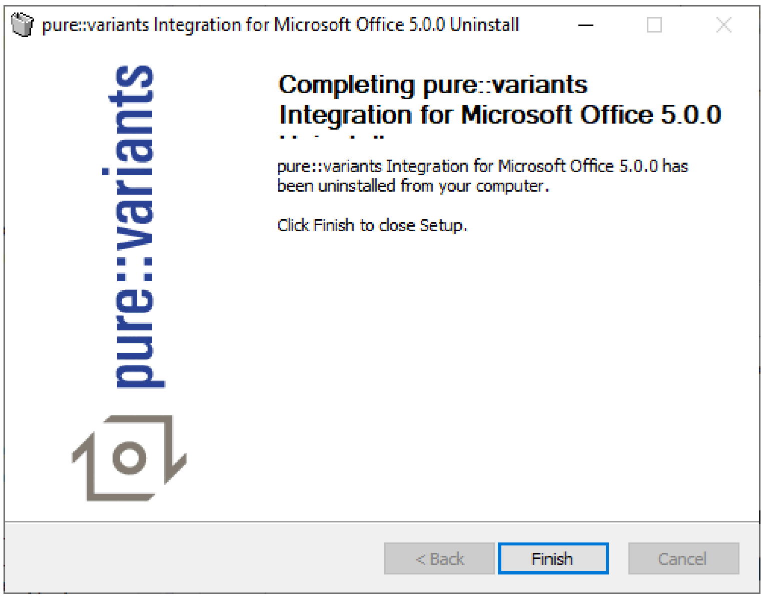
Remove pure::variants Integration for Microsoft Office 5.0.0 from your computer.

pure::variants Integration for Microsoft Office 5.0.0 will be uninstalled from the following folder. Click Uninstall to start the uninstallation.

Uninstalling from: C:\Program Files\pure-systems\pv_Enterprise_5.0\com.ps.consul.eclips

Click *Uninstall* to start the uninstall process.

Figure 90. Completing Uninstall


pure::variants Integration for Microsoft Office 5.0.0 Uninstall

Completing pure::variants Integration for Microsoft Office 5.0.0 Uninstall

pure::variants Integration for Microsoft Office 5.0.0 has been uninstalled from your computer.

Click Finish to close Setup.

The installation is succesfully finished. Click *Finish* to close the uninstaller.

7.4. Basic Setup of pure::variants Tool Integrations

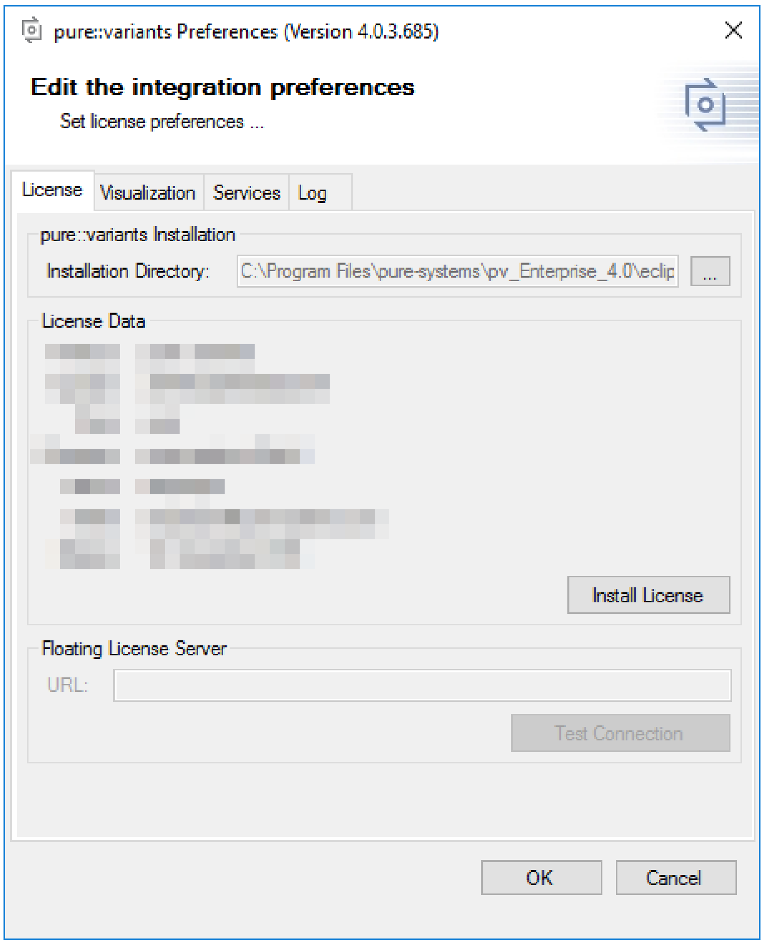
When you first use the Desktop Hub after installation, it is necessary to check whether the license preferences are correct. To this end, open the preferences dialog via the button in the Desktop Hub window or by selecting **Hub Configuration** from the pure::variants tray menu.

A dialog opens that shows the path to your pure::variants installation and your license information (see Figure 91, “Preferences Dialog”). If any of the information is missing, you need to enter it. Use the **...** button in the **pure::variants Installation** group to enter the installation directory, and the **Install License** button to specify your license.

If you are using a floating license and the URL in the **Floating License Server** group is not set already, you need to enter the URL. To test if the connection to the floating license server is established, press the button **Test Connection**.

Now you can use the Desktop Hub.

Figure 91. Preferences Dialog



pure::variants Preferences (Version 4.0.3.685)

Edit the integration preferences

Set license preferences ...

pure::variants Installation

Installation Directory: C:\Program Files\pure-systems\pv_Enterprise_4.0\eclip

License Data

Floating License Server

URL:

7.4.1. Server Connection Setup

If you are using a pure::variants floating license, establishing a connection with the pure::variants license server may need extra configuration. This may be the case, for example, if the license server is located behind a proxy server or the communication with the server is encrypted and a self-signed certificate is used.

Since pure::variants 4.0.19, the way the connection to pure::variants model or license servers is done has changed. Therefore, no advanced setup should be necessary anymore, since the settings configured in Windows are used (e.g., proxy settings, certificates). However, it is still possible to switch back to the previous server connection behavior.

Switching Back to Previous Server Connection Behaviour

This sections describe the switch back to the Java based Soap connection method. It is not recommended to use this but may help to fix connection problems. This mechanism is available for .net based integrations only.

.net based integrations are

- Section 7.1.2, "pure::variants Integration for Doors"
- pure::variants Integration for Microsoft Office
- Section 7.1.5, "pure::variants Integration for Enterprise Architect"
- Section 7.1.7, "pure::variants Integration for Team Foundation Server"

There are two ways to switch back to the previous server connection behaviour. Either you add the Windows environment variable `PV_FORCE_JAVA_SOAP_SERVER_CONNECTION` with value `true`, or you add line `forceJavaSoapConnection=true` to file `pv.properties` (see below for instructions how to edit `pv.properties`).

Once you have switched back, you may again need extra configuration to connect to a license or model server. See the section called “Advanced Integration Setup” for extra configuration setps.

Advanced Integration Setup

This steps are necessary only, if you have switched the Soap server connection method to the ols behavior. See the section called “Switching Back to Previous Server Connection Behaviour”.

If you are using a pure::variants floating license, establishing a connection with the pure::variants license server may need extra configuration. This may be the case, for example, if the license server is located behind a proxy server or the communication with the server is encrypted and a self-signed certificate is used.

All extra configurations can be done by manually editing file `pv.properties`. You can find it in two different locations:

- If you want to configure the settings for all users on the machine, please edit

  ```
  %PROGRAMDATA%/pure-variants-5/pv.properties
  ```

- If you want to configure the settings only for the current user, please edit

  ```
  %APPDATA%/pure-variants-5/pv.properties
  ```

When editing `pv.properties`, please note that:

- In case the file does not exist, you need to create it and any necessary folders first.
- Before editing `pv.properties`, make sure that no pure::variants Integration is running (e.g. Integration for Word, Excel, Doors, or the p::v Desktop Hub), otherwise your changes may be overwritten when closing the integration.
- Path delimiters in any paths you enter must be forward slashes or escaped backward slashes (`/` or `\\`). Otherwise the path cannot be read.
- All property names are case-sensitive.

Proxy Settings

The following properties can be set to configure your proxy settings.

Table 9. Proxy Settings

Property Name	Comments based on Java system property documentation
http.proxyHost	The hostname, or address, of the proxy server
http.proxyPort	The port number of the proxy server
https.proxyHost	The hostname, or address, of the proxy server in case HTTPS is used
https.proxyPort	The port number of the proxy server in case HTTPS is used
http.nonProxyHosts	Indicates the hosts that should be accessed without going through the proxy. Typically this defines internal hosts. The value of this property is a list of hosts, separated by the '|' character. In addition the wildcard character '*' can be used for pattern matching. For example http.nonProxyHosts=*.foo.com|localhost will indicate that every hosts in the foo.com domain and the localhost should be accessed directly even if a proxy server is specified. The default value excludes all common variations of the loopback address.
java.net.useSystemProxies	Set this to "true" to use Windows' global proxy settings (default: false), which are set in the Internet Explorer or in the Windows system settings. If one of the above properties is set, it overrides the respective Windows system property.

For example to use Windows' proxy settings, you would need to append this line to `pv.properties`:

```
java.net.useSystemProxies=true
```

Or to set all properties manually, you would need to append something like this:

```
http.proxyHost=YourHTTPProxyHost
http.proxyPort=80
https.proxyHost=YourHTTPSProxyHost
https.proxyPort=443
http.nonProxyHosts=*.foo.com|localhost
```

HTTPS Connection with License Server

The following HTTPS-related properties can be set. For more details, please refer to the respective Java system property documentation.

Table 10. HTTPS Settings

Property Name	Comments
javax.net.ssl.trustStore	Path to your trust store
javax.net.ssl.trustStorePassword	Password of your trust store
javax.net.ssl.trustStoreType	Trust store type (e.g. JKS)
javax.net.ssl.keyStore	Path to your key store
javax.net.ssl.keyStorePassword	Password of your key store
javax.net.ssl.keyStoreType	Key store type (e.g. JKS)
javax.net.debug	Activation of debug mode (e.g. "all" to write all possible debug logs)
com.sun.net.ssl.checkRevocation	Enable certificate revocation checking

For example when using a self-signed certificate that is stored in trust store `D:/sandbox/servercert/cert-trusted.jks` you would need to append the following lines:

```
javax.net.ssl.trustStore=D\:/sandbox/servercert/cert-trusted.jks
javax.net.ssl.trustStorePassword=password
```

HTTPS Connection with Model Access Service (pure::variants Desktop Hub only)

Per default, the self-signed certificate that is used for securing the model access service connection is only generated for the current user. Thus, when there are multiple users working on the same machine, each user would use a different self-signed certificate and each user would get a security exception the first time he uses the model access service (e.g., when working with DoorsNG).

To prevent that, you as an administrator, can manually configure which certificate is used for all users and register it in the Windows *Trusted Root Certification Authorities* store to make sure no security exception is shown. To achieve that, you can set the following properties in

```
%PROGRAMDATA%/pure-variants-5/pv.properties
```

Table 11. Model Access Service Settings

Property Name	Comments
enableHttpService	true if the model access service should be enabled
httpServicePort	Port that the model access service should run on
enableHTTPS	true if the connection should be secured
keystore	Path to your key store
keystorePassword	Password of the given key store

For example, if you wanted to enable the model access service for all users, use a secure connection, and use your own keystore that is located at `C:/ProgramData/pure-variants-5/selfsigned.jks`, you would need to append the following lines to `pv.properties`:

```
# enable the model access service
enableHttpService=true
# secure the connection
enableHTTPS=true
# the port on which the service listens
httpServicePort=9443
# the path to your keystore, which must be a java keystore and which contains your certificate
keystore=C:/ProgramData/pure-variants-5/selfsigned.jks
# the password to the keystore
keystorePassword=password
```

8. pure::variants Web Integration

8.1. IBM Rational DOORS NG Web Integration

The pure::variants Integration for DOORS NG is distributed in a WAR archive (`com.ps.consul.web.ui.doorsng-x.x.x.war`) and can be found in the **pure::variants Windows Installer package** on the pure::variants update site.

Note

For brevity we have renamed the war-archive to **pvwidget.war**. At least for Apache Tomcat, the war-archive name implies the context-path, so the web components will be reachable at **https://[pv-server-FQDN]:[port]/pvwidget/pvscl.xmlpv** .

Remember FQDN is the fully qualified domain name. The port number might be optional, if configured with standard SSL port (443)

8.1.1. Requirements for pure::variants Integration Deployment

The deployment of pure::variants Integration for DOORS NG has the following requirements:

- Extenstion must be hosted on a web server application that can be configured to run with Oracle JDK/JRE or OpenJDK.
- Extension must be accessible from a web server via HTTPS.
- The web server must not require any form of authentication to read the extension files.
- The certificate that is installed in the web server must be a valid certificate and must match the server's domain.
- Java Runtime Envritonment (JRE) or Java Development Kit (JDK) version 1.6 or later.

8.1.2. Installation on Apache Tomcat

Software Requirements

Apache Tomcat 8.5.x is recommended for deployment.

Installation of the pure::variants Integration for DOORS NG

The pure::variants Integration for DOORS NG must be deployed on your application server. This has to be done once by the system administrator. It entails copying of ***pvwidget.war*** into the ***webapps*** directory, which is located in the Apache Tomcat installation directory.

Update or reinstallation of pure::variants Integration for DOORS NG

Stop the Apache tomcat. Go to the Apache Tomcat installation directory, go in to **work** directory then go in to **Catalina** directory and delete the **localhost** directory. Go back to Apache Tomcat installation directory, go in to

webapps directory and **delete** the previously installed **pvwidget.war** and **pvwidget** directory. Now **copy** the new **pvwidget.war** in the **webapps** directory and start the Apache Tomcat again. No configuration change in Apache Tomcat is required in case of an update.

8.1.3. Installation on Websphere Liberty

Software Requirements

WebSphere Liberty Kernel v19.0.0.6+ is recommended for deployment.

Server Setup

Please follow the following steps for Websphere Liberty setup:

1. Install the WebSphere Liberty according to the official documentation.
2. Create a server by running the following command:

```
bin\server create [SERVER_NAME]
```

3. Please add the following lines into the **server.config.dir/server.xml** file:

```
<featureManager>
  <feature>jsp-2.3</feature>
  <feature>ssl-1.0</feature>
</featureManager>
```

   Please run the following command:

```
bin\installUtility install [SERVER_NAME]
```

SSL Configuration

Please ensure that the server is configured with SSL. Please refer to the **Securing communications with Liberty** section of the official documentation.

Installation of the Web Components

Copy the war archive file **pvwidget.war** to **server.config.dir/apps**. Add following line into the **server.config.dir/server.xml** file:

```
<webApplication contextRoot="pvwidget" location="${server.config.dir}/apps/pvwidget.war"/>
```

The final **server.config.dir/server.xml** file should look similar to this:

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
  <!-- Enable features -->
  <featureManager>
    <feature>jsp-2.2</feature>
    <feature>ssl-1.0</feature>
  </featureManager>

  <httpEndpoint host="*" httpPort="-1" httpsPort="9443" id="defaultHttpEndpoint"/>
  <ssl id="defaultSSLConfig" keyStoreRef="defaultKeyStore" sslProtocol="SSL"/>
  <keyStore id="defaultKeyStore" location="keystore.jks" password="{xor}LykrOiwr"
 type="JCEKS"/>

  <webApplication contextRoot="pvwidget" location="${server.config.dir}/apps/pvwidget.war"/>
</server>
```

Update or reinstallation of the pure::variants Integration for DOORS NG

Stop the WebSphere Liberty Server. Replace the **pvwidget.war** in the **server.config.dir/apps** directory. **Start** the web server again.

8.1.4. Uninstall the pure::variants Integration for Doors NG

Uninstall on Apache Tomcat

In order to uninstall pure::variants web components, stop the Apache Tomcat. Please go to Apache Tomcat installation directory and then go to **webapps** directory. Delete the **pvwidget.war** file and the **pvwidget** directory from the webapps folder.

Go back to Apache Tomcat installation directory, go in to **work** directory and then go in to **Catalina** directory and delete the **localhost** directory from it. Start the Apache Tomcat.

Uninstall on Websphere Liberty

In order to uninstall the pure::variants web components, please stop the WebSphere Liberty server. Go to **server.config.dir/apps** folder and remove the**pvwidget.war**. Start the WebSphere Liberty server again.

8.1.5. Basic Setup of the pure::variants Integration for Doors NG

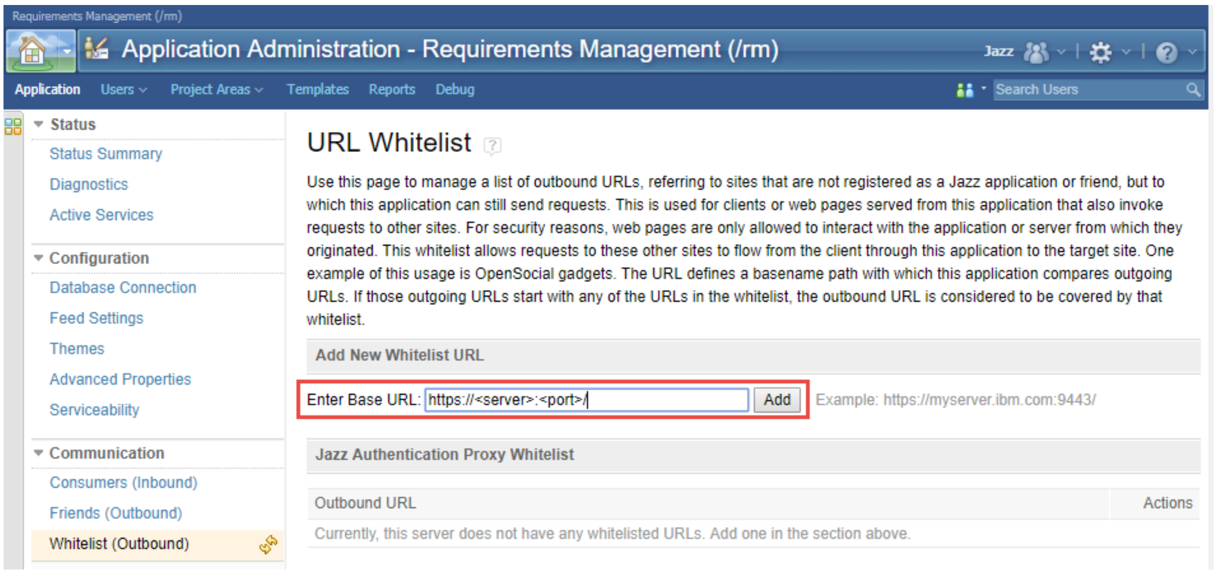
Setup on JTS/RM (Administrator)

In order to visualize the variability, few settings need to be configured on the Jazz side.

Jazz RM Whitelist

As shown in Figure 92, "Adding A URL In RM Whitelist", go to Jazz RM application administration page. Click the **Whitelist (Outbound)** on the left menu. In the **Enter Base URL** field of the **Add New Whitelist URL** section, enter the **Base URL** of the server where the pure::variants Integration for DOORS NG is installed. Click **Add** button and the newly added URL should be added to the **Outbound URL** list on the same page.

Figure 92. Adding A URL In RM Whitelist



URL Whitelist

Use this page to manage a list of outbound URLs, referring to sites that are not registered as a Jazz application or friend, but to which this application can still send requests. This is used for clients or web pages served from this application that also invoke requests to other sites. For security reasons, web pages are only allowed to interact with the application or server from which they originated. This whitelist allows requests to these other sites to flow from the client through this application to the target site. One example of this usage is OpenSocial gadgets. The URL defines a basename path with which this application compares outgoing URLs. If those outgoing URLs start with any of the URLs in the whitelist, the outbound URL is considered to be covered by that whitelist.

Add New Whitelist URL

Enter Base URL: https://<server>:<port>/ Add Example: https://myserver.ibm.com:9443/

Jazz Authentication Proxy Whitelist

Outbound URL	Actions
Currently, this server does not have any whitelisted URLs. Add one in the section above.	

Note

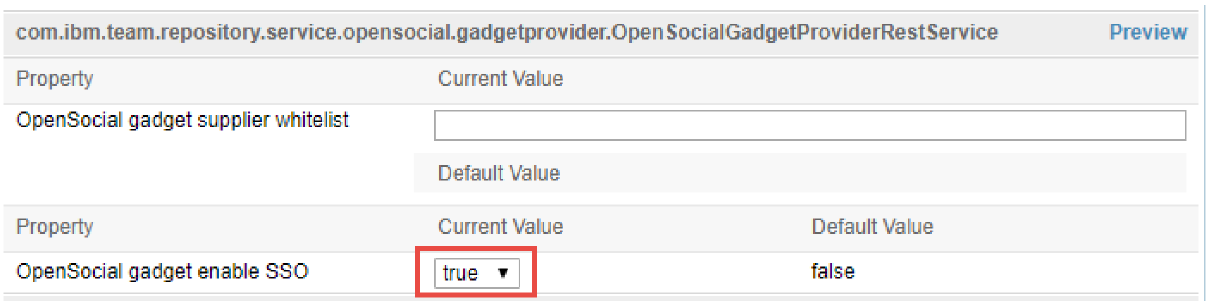
On clicking the **Add** button, Jazz tries to access the URL being added. And if the URL is accessible only then the URL is added to the list of **Outbound URL**.

JTS Advanced Settings

Browse to JTS home page. Click **Manager Server** and then click **Advanced Properties** in the left menu.

Scroll down to **OpenSocial gadget enable SSO** and set its value to **true** as shown in figure Figure 93, “OpenSocial gadget enable SSO Setting”.

Figure 93. OpenSocial gadget enable SSO Setting

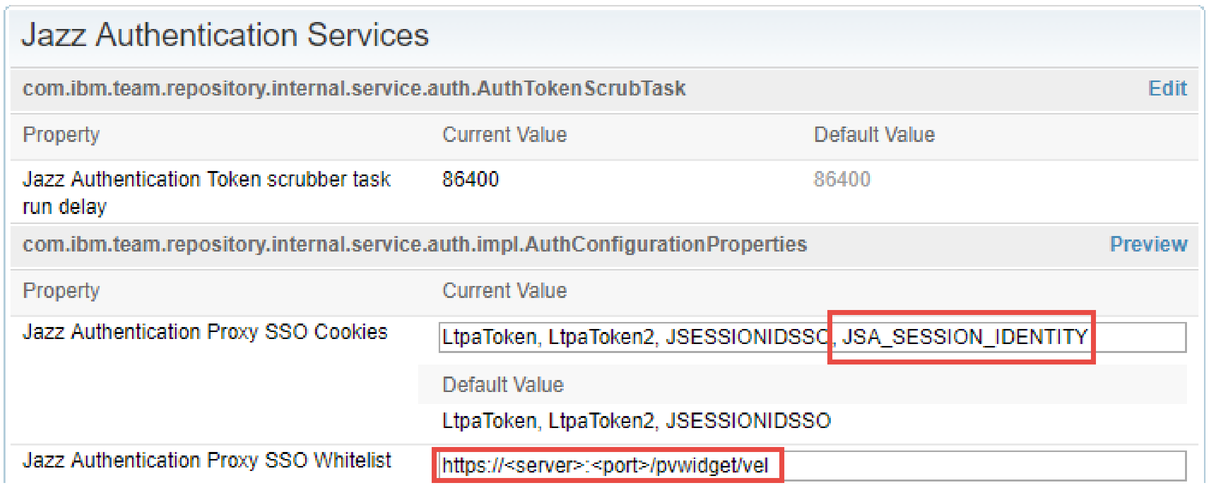


com.ibm.team.repository.service.opensocial.gadgetprovider.OpenSocialGadgetProviderRestService Preview

Property	Current Value	Default Value
OpenSocial gadget supplier whitelist		
OpenSocial gadget enable SSO	true	false

Further, scroll down to **Jazz Authentication Services**, in **Jazz Authentication Proxy SSO Cookies** field and replace its value with `LtpaToken, LtpaToken2, JSESSIONIDSSO, JSA_SESSION_IDENTITY`. In the next field in **Jazz Authentication Proxy SSO Whitelist**, write the URL of the server hosting the widget. Assuming that pure::variants Integration DOORS NG `war` was renamed as `pvwidget.war`, the URL must end with `/pvwidget/vel` as shown in the figure Figure 94, “Jazz Authentication Proxy SSO Settings”.

Figure 94. Jazz Authentication Proxy SSO Settings



Jazz Authentication Services

com.ibm.team.repository.internal.service.auth.AuthTokenScrubTask Edit

Property	Current Value	Default Value
Jazz Authentication Token scrubber task run delay	86400	86400

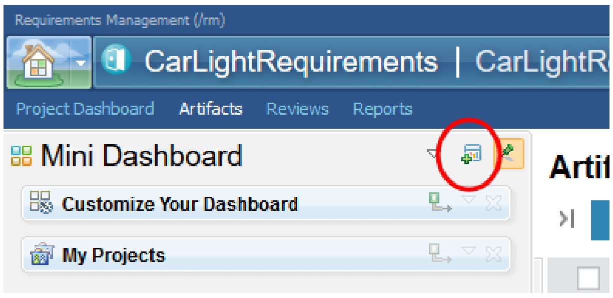
com.ibm.team.repository.internal.service.auth.impl.AuthConfigurationProperties Preview

Property	Current Value	Default Value
Jazz Authentication Proxy SSO Cookies	LtpaToken, LtpaToken2, JSESSIONIDSSO, JSA_SESSION_IDENTITY	LtpaToken, LtpaToken2, JSESSIONIDSSO
Jazz Authentication Proxy SSO Whitelist	https://<server>:<port>/pvwidget/vel	

8.1.6. Add pure::variants Integration to Doors NG

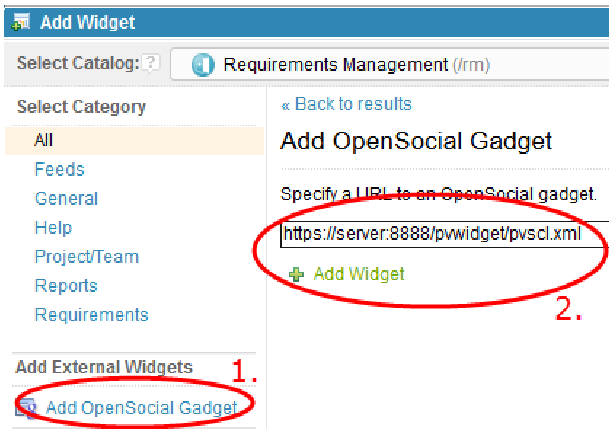
In order to interact with the Integration, it needs to be added inside Jazz' *Mini Dashboard*. Hence, open the *Mini Dashboard* on the left side panel inside DOORS NG. At the top left of Mini Dashboard, click the **Add Widget** button.

Figure 95. Adding Integration inside Mini Dashboard



A new window will popup, inside that window navigate to the **Add OpenSocial Gadget**, and enter the *URL*, wich points to the Integration. The URL should have the following pattern: *https://[server]:[port]/pvwidget/pvscl.xml*

Figure 96. Entering the URL to Integration



For further instructions see section *Adding an OpenSocial gadget* in the documentation of IBM's *Rational Collaborative Lifecycle Management*

Note

If **pure::variants Integration for DOORS NG** is going to be used in **Web Hub** mode with **Jazz V7.0** then it is required that the pure::variants Web Components' FQDN must be same as DOORS NG's FQDN. For example, if DOORS NG is accessible via https://**company.local.net**:9443/rm then pure::variants Web Components must be accessible via URL like https://**company.local.net**:8443/pv

9. pure::variants Web Components

9.1. Installation

The pure::variants Web Components are distributed in a WAR archive (`com.ps.consul.server.oslc-x.x.x.war`) and can be found in the **pure::variants Windows Installer package** on the pure::variants update site.

9.1.1. Install pure::variants Web Components

In this document, installation with Apache Tomcat and WebSphere Liberty is explained.

Prerequisite

The installation of the Web Components requires **Oracle JDK/JRE** or **OpenJDK** as the Java runtime. Any web application server that can be configured to run with the above mentioned Java runtimes can be used to host the Web Components.

Configuration of FQDN

For brevity, the war-archive is renamed to **pv.war**. In Apache Tomcat, the war-archive name implies the context-path, so the Web Components will be reachable at **https://[pv-server-FQDN]:[port]/pv** .

Remember FQDN is the fully qualified domain name. The port number is optional, if configured with standard SSL port (443).

Note

If **pure::variants Integration for DOORS NG** is going to be used in **Web Hub** mode with **Jazz V7.0** then it is required that the pure::variants Web Components' FQDN must be same as DOORS NG's FQDN. For example, if DOORS NG is accessible via https://**company.local.net**:9443/rm then pure::variants Web Components must be accessible via URL like https://**company.local.net**:8443/pv

Configuration Directory and License File

The Web Components requires access to a directory location (hereinafer referred to as Configuration Directory) where it has write permissions.

A license file issued by pure-systems is required to use the Web Components. This license file has to be stored in the **Configuration Directory**.

Note

The Configuration Directory must be persisted over the system restarts.

Installation on Apache Tomcat

On a Windows System, Apache Tomcat can be installed as a Windows Service or can be used without installation just by extracting the Apache Tomcat archive. In the following sections, configuration for both kind of setups are explained.

Software Requirements

Apache Tomcat 8.5.x is recommended for deployment.

SSL Configuration

Enable the **Apache Tomcat** to use **SSL** either by creating a self-signed certificate or by acquiring a certificate that is signed by a trusted certification authority (CA). Please see the official documentation of Apache Tomcat.

Apache Tomcat Installation as a Windows Service

Apache Tomcat Installation

Download the Apache Tomcat's Windows Service Installer. Run the installer to install the Tomcat as a Windows Service.

Install the pure::variants Web Components

Copy the pure::variants Web Components war **pv.war** into the `webapps` directory in the Apache Tomcat's installation directory.

Configuration of Apache Tomcat

Double click the Apache Tomcat icon on the Windows Taskbar to open the Apache Tomcat Service Properties dialog.

Figure 97. Apache Tomcat Service Icon on Windows Taskbar

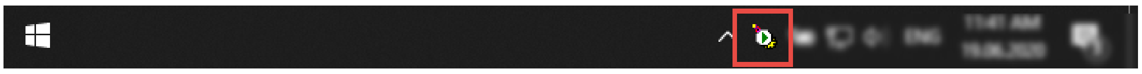
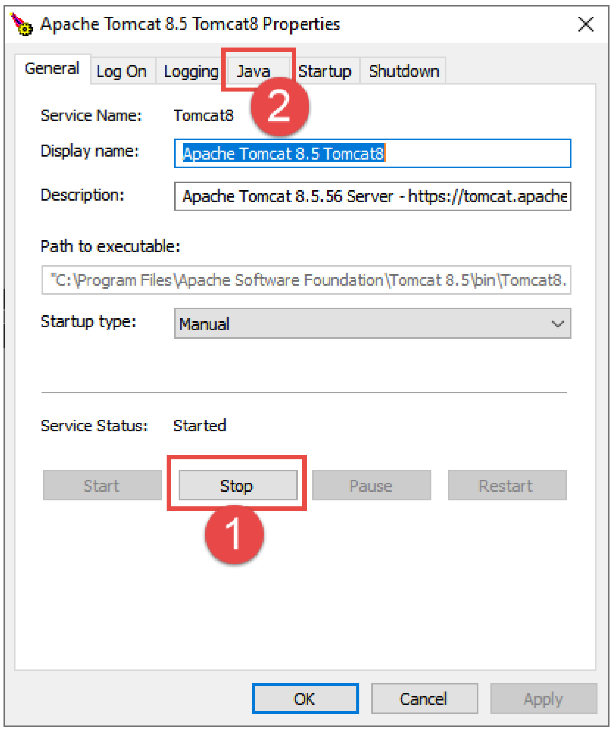


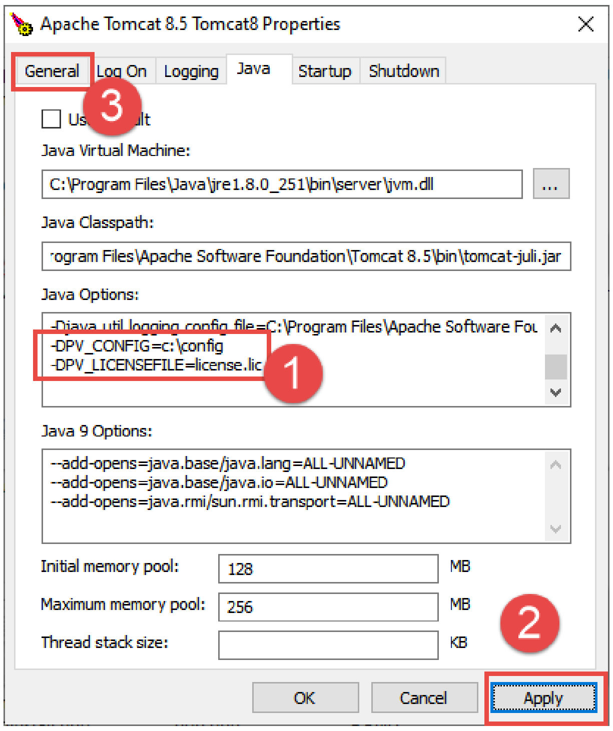
Figure 98. Apache Tomcat Service Properties



Service Name: Tomcat8
Display name: Apache Tomcat 8.5 Tomcat8
Description: Apache Tomcat 8.5.56 Server - https://tomcat.apache
Path to executable:
"C:\Program Files\Apache Software Foundation\Tomcat 8.5\bin\Tomcat8.
Startup type: Manual
Service Status: Started

Click the **Stop** button to stop the Tomcat service. Click on **Java** Tab.

Figure 99. Apache Tomcat Service - Java Tab



Java Virtual Machine:
C:\Program Files\Java\jre1.8.0_251\bin\server\jvm.dll
Java Classpath:
'ogram Files\Apache Software Foundation\Tomcat 8.5\bin\tomcat-juli.jar
Java Options:
-Djava.util.logging.config.file=C:\Program Files\Apache Software Fou
-DPV_CONFIG=c:\config
-DPV_LICENSEFILE=license.lic
Java 9 Options:
--add-opens=java.base/java.lang=ALL-UNNAMED
--add-opens=java.base/java.io=ALL-UNNAMED
--add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
Initial memory pool: 128 MB
Maximum memory pool: 256 MB
Thread stack size: KB

In the **Java Options** section, add the following parameters to the list.

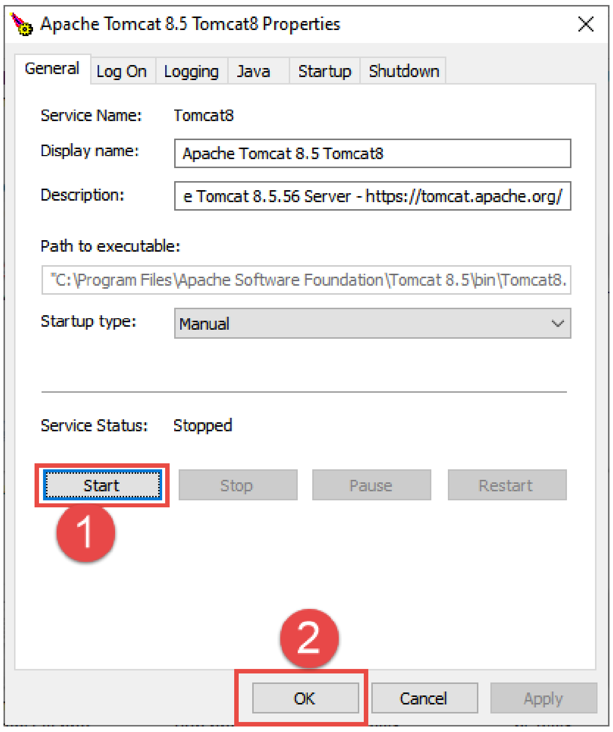
```
-DPV_CONFIG=[DIRECTORY_PATH]
-DPV_LICENSEFILE=[LICENSE_FILE_NAME]
```

Replace the DIRECTORY_PATH and LICENSE_FILE_NAME placeholders with the values corresponding to your system configuration. An example configuration could look like this:

```
-DPV_CONFIG=c:\config
-DPV_LICENSEFILE=license.lic
```

Click **Apply** to save the settings. Click the **General** tab button.

Figure 100. Apache Tomcat Service Properties


Apache Tomcat 8.5 Tomcat8 Properties

General | Log On | Logging | Java | Startup | Shutdown

Service Name: Tomcat8
Display name: Apache Tomcat 8.5 Tomcat8
Description: e Tomcat 8.5.56 Server - https://tomcat.apache.org/
Path to executable:
"C:\Program Files\Apache Software Foundation\Tomcat 8.5\bin\Tomcat8.
Startup type: Manual
Service Status: Stopped

Click the **Start** button to start the service again. Click **Ok** to close the Apache Tomcat Properties dialog.

Note

For further details, please visit the Apache Tomcat: Windows Service How-To.

Apache Tomcat Installation from Archive

Download the Apache Tomcat's archive file corresponding to your operarting system. Extract the archive in a directory.

Install the pure::variants Web Components

Copy the pure::variants Web Components war file **pv.war** into the **`webapps`** directory in the Apache Tomcat's directory.

Configuration of Apache Tomcat

Set the abolute path to Configuration Directory in the Java system property named **PV_CONFIG**. Set the name of the License File in the Java System property named **PV_LICENSEFILE**.

In case of **Windows**, create/open the file **`setenv.bat`** in the **`bin`** directory of Apache Tomcat's installation directory. Add the following and save it.

```
set "JAVA_OPTS=-DPV_CONFIG=[DIRECTORY_PATH] -DPV_LICENSEFILE=[LICENSE_FILE_NAME]"
```

Considering DIRECTORY_PATH to be `C:\config` and LICENSE_FILE_NAME as `license.lic` the configuration in setenv.bat would like this:

```
set "JAVA_OPTS=-DPV_CONFIG=C:\config -DPV_LICENSEFILE=license.lic"
```

In case of **Linux**, create/open the file `setenv.sh` in the `bin` directory of Apache Tomcat's installation directory and add the following and save it.

```
set JAVA_OPTS="-DPV_CONFIG=[DIRECTORY_PATH] -DPV_LICENSEFILE=[LICENSE_FILE_NAME]"
```

Starting the Apache Tomcat

After completing the above setup, go to the **bin** directory of the Apache Tomcat's installation directory and execute the **Startup.bat (Windows)** or **Startup.sh (Linux)**.

Install on WebSphere Liberty

Software Requirements

WebSphere Liberty Kernel v19.0.0.6+ is recommended for deployment.

Server Setup

Please follow the following steps for Websphere Liberty setup:

1. Install the WebSphere Liberty according to the official documentation.

2. Create a server by running the following command:

   ```
   bin\server create [SERVER_NAME]
   ```

3. Create the Configuration Directory and put the Web Components license file in this directory. Please create **jvm.options** file into **server.config.dir** (see Directory locations and properties) and add the following lines into:

   ```
   -DPV_CONFIG=[DIRECTORY_PATH]
   -DPV_LICENSEFILE=[LICENSE_FILE_NAME]
   ```

   In case of Windows operating system, considering DIRECTORY_PATH to be `C:\config` and LICENSE_FILE_NAME to be `license.lic`, above mentioned settings will look like this:

   ```
   -DPV_CONFIG=C:\config
   -DPV_LICENSEFILE=license.lic
   ```

4. Please add the following lines into the **server.config.dir/server.xml** file:

   ```
   <featureManager>
     <feature>jsp-2.3</feature>
     <feature>ssl-1.0</feature>
   </featureManager>
   ```

   Please run the following command:

   ```
   bin\installUtility install [SERVER_NAME]
   ```

SSL Configuration

Please ensure that the server is configured with SSL. Please refer to the **Securing communications with Liberty** section of the official documentation.

Session Configuration

Please ensure that the server is configured with a unique cookie name for session identification. Add the following line into **server.config.dir/server.xml** file:

```
<httpSession cookieName="PVSESSIONID"/>
```

Note

In case PVSESSIONID as a cookie name is already used, e.g. by a proxy server, please choose another unique name.

Install the pure::variants Web Components

Copy the pure::variants Web Components war file **pv.war** to **server.config.dir/apps**. Add following line into the **server.config.dir/server.xml** file:

```
<webApplication contextRoot="pv" location="${server.config.dir}/apps/pv.war"/>
```

The final **server.config.dir/server.xml** file should look similar to this:

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
  <!-- Enable features -->
  <featureManager>
    <feature>jsp-2.2</feature>
    <feature>ssl-1.0</feature>
  </featureManager>

  <httpEndpoint host="*" httpPort="-1" httpsPort="9443" id="defaultHttpEndpoint"/>
  <ssl id="defaultSSLConfig" keyStoreRef="defaultKeyStore" sslProtocol="SSL"/>
  <keyStore id="defaultKeyStore" location="keystore.jks" password="{xor}LykrOiwr"
 type="JCEKS"/>

  <httpSession cookieName="PVSESSIONID"/>
  <webApplication contextRoot="pv" location="${server.config.dir}/apps/pv.war"/>
</server>
```

9.1.2. Advanced Setup

Add custom images for pure::variants Web Components

To have custom images shown inside of the Model Viewer/Editor in respect to specific element types, an image directory must be configured manually to the setup, additionally. Therefore please navigate to your pv.properties file, which location is already defined by PV_CONFIG variable. Please append the following key-value pair into the pv.properties file: `com.ps.consul.oslc.serviceprovider.custom.icons.dir.path=/path/to/images`

Note

For Windows, please take care on escaping your path properly. Each occurance of the following characters must be replaced like follows: ':' -> '\:' and '\' -> '\\';

Example: *C:\config\images* => *C\:\\config\\images*

9.1.3. Update pure::variants Web Components

Update on Apache Tomcat

In order to update the pure::variants Web Components, **Stop** the Apache Tomcat. Please go to Apache Tomcat installation directory, go in to **work** directory and go further in to the **Catalina** directory and **Delete** the **localhost** directory from it. Navigate back to the Apache Tomcat installation directory and then go to the **webapps** directory. **Delete** the **pv.war** and **pv** directory from the webapps folder. Place the new **pv.war** in the **webapps** directory. **Start** the Apache Tomcat again.

Update on Websphere Liberty

In order to update the pure::variants Web Components, please **Stop** the WebSphere Liberty server. Go to **server.config.dir/apps** folder and **replace** the **pv.war** with the new pv.war. **Start** the WebSphere Liberty server again.

9.1.4. Uninstall pure::variants Web Components

Remove Web Components

Uninstall on Apache Tomcat

In order to uninstall pure::variants Web Components, stop the Apache Tomcat. Please go to Apache Tomcat installation directory and then go to **webapps** directory. Delete the **pv.war** file and the **pv** directory from the webapps folder.

Go back to Apache Tomcat installation directory, go in to **work** directory and then go in to **Catalina** directory and delete the **localhost** directory from it.

Uninstall on Websphere Liberty

In order to uninstall the pure::variants Web Components, please stop the WebSphere Liberty server. Go to **server.config.dir/apps** folder and remove the **pv.war**.

Open the **server.config.dir/server.xml** and remove the following line:

```
<webApplication contextRoot="pv" location="${server.config.dir}/apps/pv.war"/>
```

Save the file.

Remove Configuration Directory

Delete the **Configuration Directory** (mentioned in the section called “Configuration Directory and License File”) from the file system.

9.2. Basic Setup

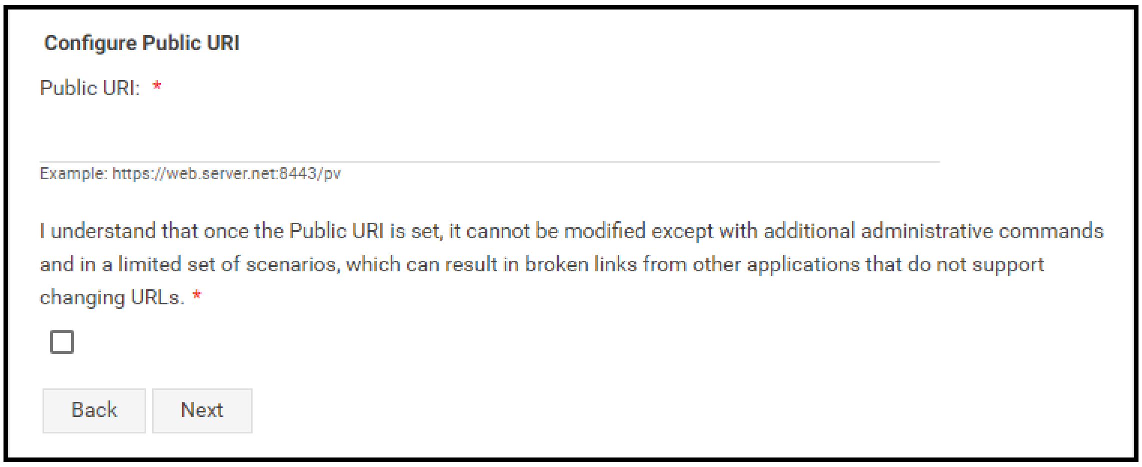
9.2.1. Setup Wizard

Start your web server hosting the pure::variants Web Components. Start the ***Setup*** by pointing the web browser to *https://[pv-server-FQDN]:[port]**/pv/ui/setup*** (assuming that the context root of the Web Components is /pv).

Public URI

Enter the **Public URI** of **pure::variants Web Components** and press **Next**.

Figure 101. The Setup Wizard, Public URI


Configure Public URI

Public URI: *

Example: https://web.server.net:8443/pv

I understand that once the Public URI is set, it cannot be modified except with additional administrative commands and in a limited set of scenarios, which can result in broken links from other applications that do not support changing URLs. *

Authentication Setup

Jazz and pure::variants Web Components can can be configured to authenticate over following methods:

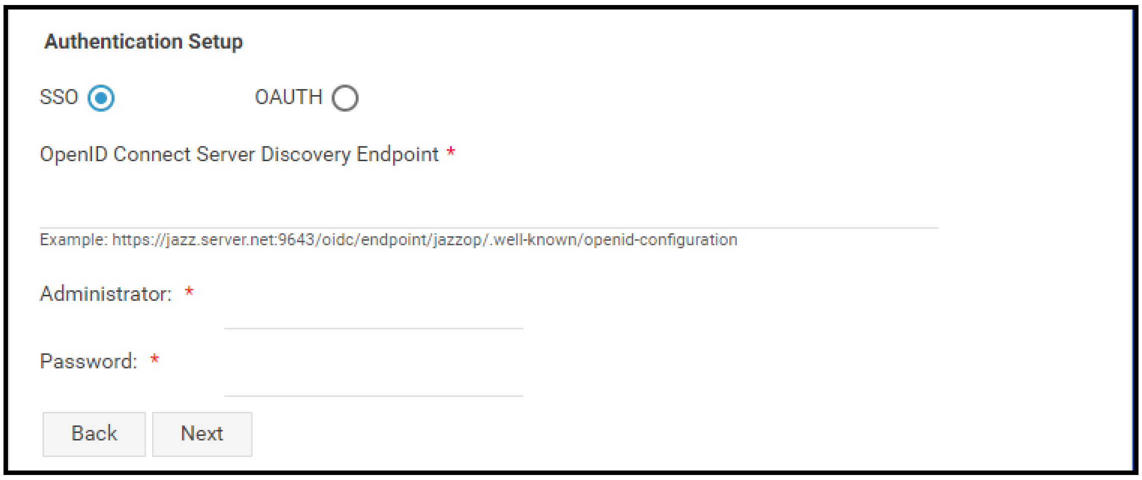
1. OpenID Connect based Single Sign On (SSO)

2. OAuth based

OpenID Connect based SSO Setup

By default *SSO* mode is selected. Enter the *OpenID Connect Server Discovery Endpoint* URL. Enter the credentials of a user with administrative rights to OIDC server. Press **Next** to continue. After successful registration with OIDC Server the setup wizard proceeds to the next step.

Figure 102. The Setup Wizard, SSO



Authentication Setup

[✓ SSO] OAUTH

OpenID Connect Server Discovery Endpoint *

Example: https://jazz.server.net:9643/oidc/endpoint/jazzop/.well-known/openid-configuration

Administrator: *

Password: *

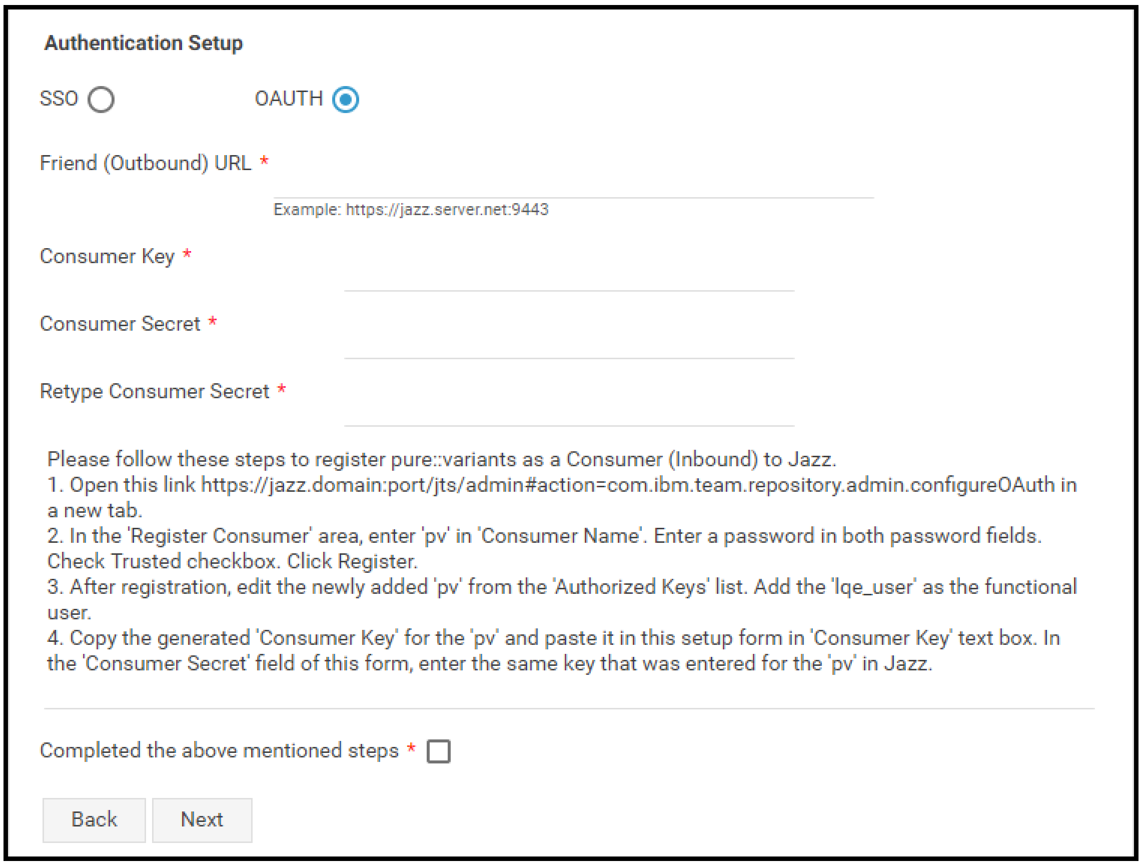
OAuth Setup

In order to setup OAuth, please select the OAuth option. Enter the **Friend (Outbound) URL**. In case of Jazz, the outbound URL would look like `https://[jazz-server-FQDN]:9443`. Enter the **Consumer Key** and type the **Consumer Secret** twice, check the check box at the bottom and click **Next** to proceed.

Note

Please follow the steps listed on the OAuth setup page to generate a pair of *Consumer Key* and *Consumer Secret*. Enter these newly generated data on the form to proceed.

Figure 103. The Setup Wizard, OAuth



Authentication Setup

SSO ○ [✓ OAUTH]

Friend (Outbound) URL *

Example: https://jazz.server.net:9443

Consumer Key *

Consumer Secret *

Retype Consumer Secret *

Please follow these steps to register pure::variants as a Consumer (Inbound) to Jazz.
1. Open this link https://jazz.domain:port/jts/admin#action=com.ibm.team.repository.admin.configureOAuth in a new tab.
2. In the 'Register Consumer' area, enter 'pv' in 'Consumer Name'. Enter a password in both password fields. Check Trusted checkbox. Click Register.
3. After registration, edit the newly added 'pv' from the 'Authorized Keys' list. Add the 'lqe_user' as the functional user.
4. Copy the generated 'Consumer Key' for the 'pv' and paste it in this setup form in 'Consumer Key' text box. In the 'Consumer Secret' field of this form, enter the same key that was entered for the 'pv' in Jazz.

Completed the above mentioned steps * ☐

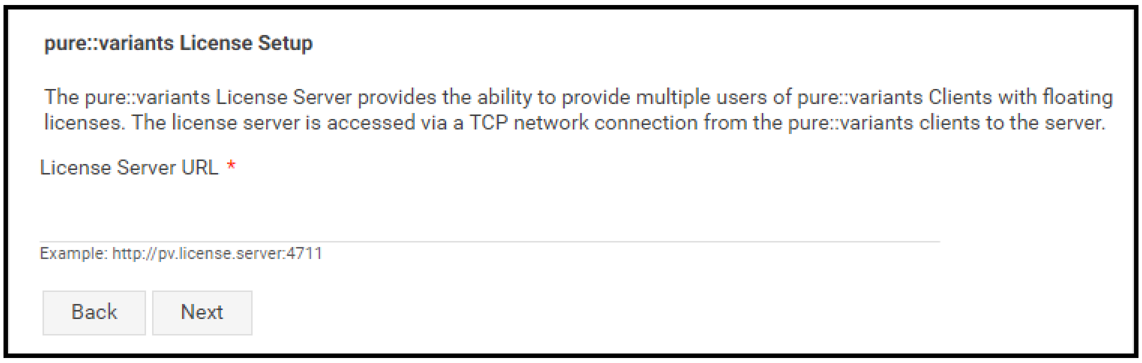
pure::variants License Server

License Server URL field is prepopulated with the URL value defined in the License File. The prepopulated value can be changed. If the field is empty then it is necessary to enter a valid License Server URL. On pressing Next the Web Components check the validity of the License Server and proceeds to the next step.

Note

In case the **License Server** is configured with **SSL**, please make sure that the pure::variants Web Components runtime is aware of the certificate with which License Server is configured.

Figure 104. The Setup Wizard, pure::variants License Server



pure::variants License Setup

The pure::variants License Server provides the ability to provide multiple users of pure::variants Clients with floating licenses. The license server is accessed via a TCP network connection from the pure::variants clients to the server.

License Server URL *

Example: http://pv.license.server:4711

pure::variants Model Server

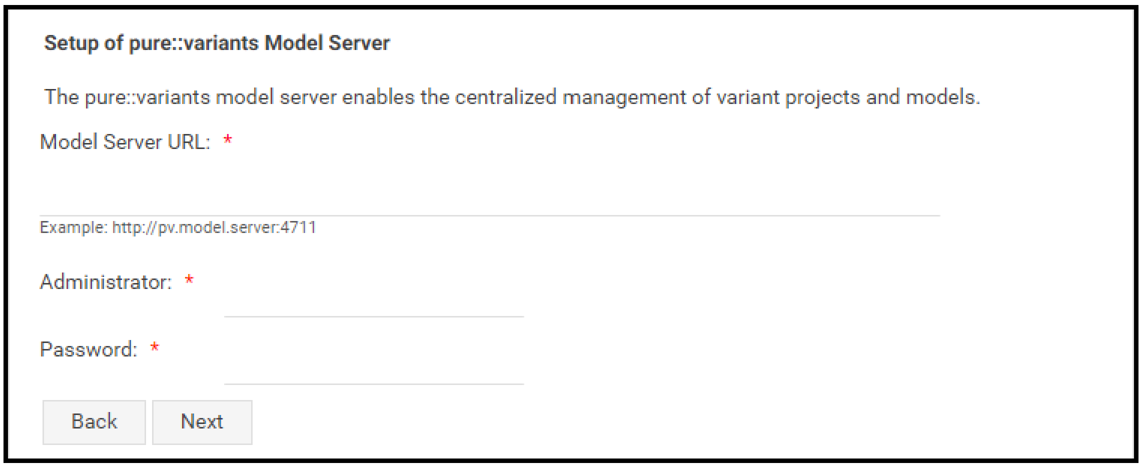
Enter the Model Server URL and the credentials of a Model Server user with the administrative privileges.

If **SSO** is enabled in the previous step then the user being entered here must have permissions to set an **alias** in the Model Server. Configuration of pure::variants Model Server provides information on how to setup an alias user. Click **Next** to proceed with the setup.

Note

In case the **Model Server** is configured with **SSL**, please make sure that the pure::variants Web Components runtime is aware of the certificate with which Model Server is configured.

Figure 105. The Setup Wizard, pure::variants Model Server



Setup of pure::variants Model Server

The pure::variants model server enables the centralized management of variant projects and models.

Model Server URL: *

Example: http://pv.model.server:4711

Administrator: *

Password: *

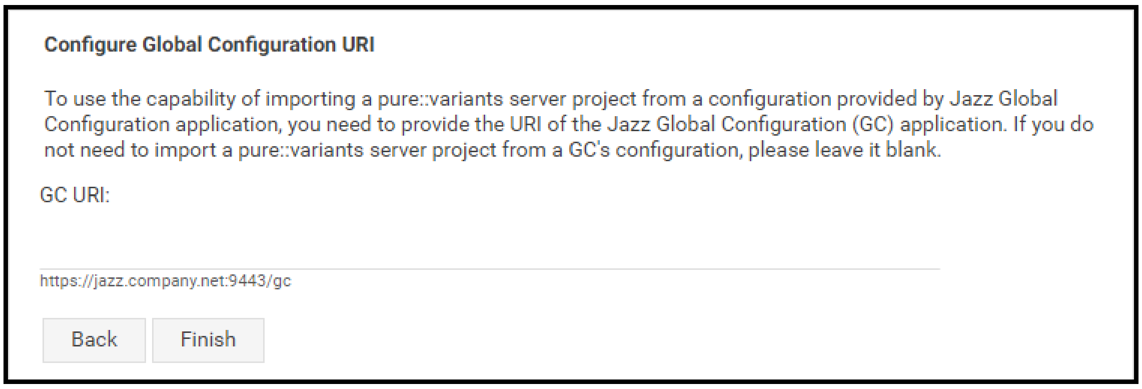
pure::variants Model Server Setup with Alias User

In order to run Model Server in an SSO enabled environment, an alias user needs to be set up. Alias user is set through a command line parameter named **aliasuser** . An example is shown below considering pvtest as the alias user: `variantsd.exe /aliasuser pvtest` .

Global Configuration URI

Please enter the Global Configuration URI in this format **https://[jazz-server-FQDN]:9443/gc**. If you don't want to configure Global Configuration URI then leave it blank. On clicking **Finish**, pure::variants web components check the validity of GC URI (if entered) and proceeds to the finish page.

Figure 106. The Setup Wizard, Global Configuration URI



Configure Global Configuration URI

To use the capability of importing a pure::variants server project from a configuration provided by Jazz Global Configuration application, you need to provide the URI of the Jazz Global Configuration (GC) application. If you do not need to import a pure::variants server project from a GC's configuration, please leave it blank.

GC URI:

https://jazz.company.net:9443/gc

Finish Page

Finish page is shown on successful completion of all steps. Please **restart** the **Web Server** hosting the Web Components.

9.2.2. Friend Setup

You need to add **pure::variants Web Components** as a **Friend** to **Jazz Team Server (JTS)** in order to enable JTS to access the pure::variants web components data. Jazz Official Documentation can be visited for details.

Note

After finishing the pure::variants web components setup and before starting the friend setup in JTS, it is a must that the web server hosting the pure::variants web components must be restarted.

OAuth mode

Procedure

1. Browse to the **JTS Admin page**. Then click **Manage Server** and then click **Friends (Outbound)**. This page can also be directly accessed by pointing your web browser to **https://[jazz-server-FQDN]:9443/jts/admin#action=com.ibm.team.repository.admin.friends**.
2. On the Friends Page, in the **Friends List** section, click **Add**.
3. In the **Add Friend** dialog, provide the following information.
   - Provide the URI for root services of the pure::variants Web Components in this form: **https://[pv-server-FQDN]:[port]/pv/rootservices**.
   - Enter a name to identify the pure::variants web components e.g. VM.
   - Click Next to continue.
4. On this dialog, perform following actions.
   - In the **OAuth Secret** and **Re-type Secret** fields, enter the OAuth secret key.
   - Select the **Trusted** check box.
   - Click **Create Friend**, and then click **Next**.
5. On the **Authorize Provisional Key** dialog, click **Grant access for the provisional key**. Perform the following steps:
   - **Login** to pure::variants server with a user with administrative privileges.
   - On the next dialog, check **Trusted** and press **Allow**.
6. Press **Finish** to complete the setup.

SSO mode

Procedure

1. Browse to the **JTS Admin page**. Then click **Manage Server** and then click **Friends (Outbound)**. This page can also be directly accessed by pointing your web browser to **https://[jazz-server-FQDN]:9443/jts/admin#action=com.ibm.team.repository.admin.friends**.
2. On the Friends Page, in the **Friends List** section, click **Add**.
3. In the **Add Friend** dialog, provide the following information.
   - Provide the URI for root services of the pure::variants Web Components in this form: **https://[pv-server-FQDN]:[port]/pv/rootservices**.
   - Enter a name to identify the pure::variants web components e.g. VM.

- Click Next to continue.

4. As both jazz and pure::variants web components are configured in SSO mode that's why this dialog does not ask for any credentials. Click **Create Friend** and click **Finish** to close the wizard.

   Note

   On step 4 of the above procedure, if you are asked to enter OAuth credentials then it is a problem. It means that JTS is unable to recognize that the pure::variants web components is also SSO enabled and registered with the same Jazz Authorization Server as JTS. In this case you have to close the friend setup dialog and wait for 10 to 15 mintues and start the friend setup again. If the problem persists then you have to restart the jazz server and then do the friend setup again.

9.3. Troubleshooting

9.3.1. Re-run Setup

In case of the setup failure due to any reason, the following steps are required to restart the setup.

i. Stop the web server hosting pure::variants web components

ii. Delete the pv.properties file. The file is located in the directory set through PV_CONFIG in JAVA_OPTS

iii. Start the web server hosting pure::variants web components

iv. Navigate to the setup page: *https://[pv-server-FQDN]:[port]/pv/ui/setup*

For information about the individual Setup steps, refer to the section Section 9.2.1, “Setup Wizard”.