



**PureWeb® STK 3.1**

Server Administration Guide

**The information contained herein is proprietary and confidential and cannot be disclosed or duplicated without the prior written consent of Calgary Scientific Inc.**

Copyright © 2012 Calgary Scientific Inc. All rights reserved.

### **About Calgary Scientific**

Calgary Scientific Inc., is dedicated to providing advanced web-enablement, mobility enhancement and advanced visualization solutions to industries looking for secure access and use of their data or graphics intensive applications, while using their existing systems. Visit [www.calgaryscientific.com](http://www.calgaryscientific.com) for more information.

### **Notice**

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Calgary Scientific Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

### **Your Responsibility for Your System's Security**

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Calgary Scientific products.

### **Trademarks**

ResolutionMD, PureWeb, the Calgary Scientific logo, and the PureWeb logo are trademarks and/or registered trademarks of Calgary Scientific Inc. or its subsidiaries. All other trademarks and trade names referred to in this document are the property of other companies.

### **Technical Support from Calgary Scientific**

Please contact Calgary Scientific Support at the regional numbers provided on [page 9](#).

### **Ordering and Licensing Information**

The PureWeb license is an ASCII text file with a .lic extension. The license is not part of the PureWeb Server installation, and must be acquired from Calgary Scientific Inc. When you receive your license, copy the .lic file in to the C:\CSI\PureWeb\Server\conf\ directory and (re)start the server.

### **Released by**

Calgary Scientific Inc. [www.calgaryscientific.com](http://www.calgaryscientific.com)

**Document Version:** PW3.1\_Server\_Admin\_Guide-08-2012\_v3.100.00

# Table of Contents

List of Tasks	.....	<b>5</b>
List of Procedures	.....	<b>6</b>
Preface	Intended Audience.....	8
	Reading Recommendations .....	8
	Making Comments on This Document .....	9
	Contacting Calgary Scientific Support .....	9
	Common Task.....	9
Chapter 1	<b>Introducing PureWeb®</b> .....	<b>13</b>
	About the PureWeb® Server .....	13
	Basic Architecture.....	13
	Prerequisites.....	15
	PureWeb® Server Installation .....	15
	PureWeb® Server License.....	15
Chapter 2	<b>Using the PureWeb® Server</b> .....	<b>16</b>
	Start the PureWeb® Server .....	16
	Stop the PureWeb® Server .....	17
	Log in to the PureWeb® Server .....	17
	Log out of the PureWeb® Server .....	18
	Monitor the PureWeb® Server .....	18
	Administrative Pages .....	19
	Manage the PureWeb® Server .....	26
	Cluster Management .....	26
	Configuration Management .....	29
Chapter 3	<b>Configuring the PureWeb® Server</b> .....	<b>40</b>
	Configuration Hierarchy .....	41
	Configuration Properties.....	42

	HTTP/HTTPS Connection Configuration.....	46
	Security Configuration .....	48
	Authentication Options.....	48
	Security Roles.....	50
	LDAP Authentication Configuration .....	53
	Database Authentication Configuration .....	55
	Reverse Proxies and Load Balancers .....	57
	Requirements for Reverse Proxies .....	57
	Recommendations for Load Balancers.....	58
Chapter 4	<b>Adding Applications to the PureWeb® Server .....</b>	<b>59</b>
	Application Files .....	59
Appendix A	<b>Secure Socket Layer (SSL) Certificate .....</b>	<b>72</b>
	About SSL Certificates .....	72
Supplements	<b>Related Documentation Resources .....</b>	<b>74</b>
	<b>Document Conventions .....</b>	<b>76</b>
Index	.....	78



## List of Tasks

Operational .....	9
Configuration .....	10
Cluster/Node Configuration .....	11
Application Installation .....	12
Request/Install SSL Certificates .....	12

# List of Procedures

Starting the PureWeb® Server .....	16
Stopping the PureWeb® Server .....	17
Logging in to the PureWeb® Server .....	17
Logging out of the PureWeb® Server .....	18
Accessing the Server Administrative Pages .....	19
Accessing Cluster Status Information .....	21
Accessing Configuration Files .....	22
Accessing Server Version Information .....	22
Accessing Application Availability Information .....	23
Accessing Support Information .....	24
Accessing License Details .....	24
Accessing Log Files .....	25
Inactivating a Clustered/Active Node .....	27
Testing a Node .....	27
Terminating an Active Rendering Session .....	28
Reloading a Plugin .....	30
Configuring an Application Plugin .....	31
Configuring Clusters .....	32
Making a Service Available Across Domains .....	34
Configuring Log File Retention Period .....	36
Changing the Administrator Password .....	37
Backing Up the PureWeb® Server .....	38
Restoring the PureWeb® Server from Backup .....	38
Changing Server Support Information .....	39
Configuring Display Devices .....	42
Configuring Application and Server Interactions .....	43
Configuring Client and Server Interactions .....	44
Disabling an HTTP Connector .....	46
Enabling an HTTPS Connector .....	47

Configuring the Authentication Type . . . . .	49
Controlling Access to Applications . . . . .	50
Adding PureWeb Users . . . . .	51
Changing Passwords . . . . .	51
Generating an Encoded Password. . . . .	52
Changing a Users Role . . . . .	52
Disabling a User . . . . .	53
Configuring LDAP Authentication. . . . .	53
Registering the LDAP Authentication Provider. . . . .	55
Configuring Database Authentication. . . . .	55
Registering the Database Authentication Provider. . . . .	56
Registering your Application. . . . .	60
Configuring your Application. . . . .	66
Adding the Resources for your Application . . . . .	67
Editing the security-config.xml File to Secure Resources . . . . .	67
Creating Private Application Resources. . . . .	68
Allocating the Graphics Processing Unit (GPU) for your Application . . . . .	68
Displaying Links to Applications on the App Page . . . . .	69
Requesting and Installing an SSL Certificate . . . . .	73

## Preface

Welcome to the *PureWeb 3.1 Server Administration Guide*. Although every effort has been made to make this document clear and easy to understand, it is still very technical in nature.

Wherever possible, cross-referencing among chapters and other documents within the PureWeb® Software Transformation Kit (STK) documentation suite has been used.

This document is valid for all 3.x release(s) of this product.

This preface contains the following sections:

- [Intended Audience, page 8](#)
- [Reading Recommendations, page 8](#)
- [Making Comments on This Document, page 9](#)
- [Contacting Calgary Scientific Support, page 9](#)
- [Common Task, page 9](#)

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on [page 74](#).

---

## Intended Audience

This document is intended to be read by software developers/server administrators that plan to install and use the PureWeb® STK.

It is assumed that the developer/server administrator has:

- A general understanding of server configuration and installation.
- A general knowledge of application development.

---

## Reading Recommendations

This preface includes tasks-based tables that contain all of the tasks and procedures needed to successfully operate and configure the PureWeb® Server.



---

## Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to [Techpubs@calgaryscientific.com](mailto:Techpubs@calgaryscientific.com).

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the scope of this document only and to the way in which the information is presented.

When you send us comments, you grant Calgary Scientific Inc. a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

---

## Contacting Calgary Scientific Support

Contact Calgary Scientific Support during the hours of 9:00AM - 5:00PM, Monday - Friday (Mountain Standard Time).

Web Site	E-Mail
<a href="http://support.getpureweb.com">http://support.getpureweb.com</a>	<a href="mailto:support@getpureweb.com">support@getpureweb.com</a>

---

## Common Task

Common operational tasks and procedures are listed in the table below.

### Task Summary: Operational

Task	Procedure
Ensure proper licensing.	The PureWeb® Server will start, but will <i>not</i> be operational without a valid license. The license is an ASCII text file with a .lic extension and is not part of the PureWeb® Server installation. To make the PureWeb® Server operational, you must acquire the license from Calgary Scientific Inc. and follow the steps in the Installing the License procedure found in the <i>PureWeb® Installation Guide for Microsoft Windows</i> .  Follow the steps in the Verify the License File is Installed procedure found in the <i>PureWeb® Installation Guide for Microsoft Windows</i> , to confirm the installation of your PureWeb® license.
Start the PureWeb® Server.	“Starting the PureWeb® Server” on <a href="#">page 16</a> .

## Task Summary: Operational (Continued)

Task	Procedure
Stop the PureWeb® Server.	“Stopping the PureWeb® Server” on <a href="#">page 17</a> .
Log in to the PureWeb® Server.	“Logging in to the PureWeb® Server” on <a href="#">page 17</a> .
Access the PureWeb® Server administrative pages.	“Accessing the Server Administrative Pages” on <a href="#">page 19</a> .
Access the PureWeb® Server version information.	“Accessing Server Version Information” on <a href="#">page 22</a> .
Access application availability information.	“Accessing Application Availability Information” on <a href="#">page 23</a> .
Access your PureWeb® Server license details.	“Accessing License Details” on <a href="#">page 24</a> .
Access cluster/node information.	<ul style="list-style-type: none"> <li>• “Accessing Cluster Status Information” on <a href="#">page 21</a>.</li> <li>• For more information on managing clusters/nodes refer to <a href="#">Task Summary: Cluster/Node Configuration</a>.</li> </ul>
Access your PureWeb® Server log files.	<ul style="list-style-type: none"> <li>• “Accessing Log Files” on <a href="#">page 25</a>.</li> <li>• To modify the log file retention period refer to “Configuring Log File Retention Period” on <a href="#">page 36</a>.</li> </ul>
Access your PureWeb® Server configuration files.	<p>“Accessing Configuration Files” on <a href="#">page 22</a>.</p> <p>From the Configuration page you can perform the following procedures:</p> <ul style="list-style-type: none"> <li>• “Configuring an Application Plugin” on <a href="#">page 31</a>.</li> <li>• “Changing the Administrator Password” on <a href="#">page 37</a>.</li> <li>• “Backing Up the PureWeb® Server” on <a href="#">page 38</a>.</li> <li>• “Restoring the PureWeb® Server from Backup” on <a href="#">page 38</a>.</li> <li>• “Changing Server Support Information” on <a href="#">page 39</a>.</li> </ul>

Common configuration tasks and procedures are listed in the table below.

## Task Summary: Configuration

Task	Procedure
Restrict access to the server.	“Adding PureWeb Users” on <a href="#">page 51</a> .
Configure the authentication type used by the server.	“Configuring the Authentication Type” on <a href="#">page 49</a> .

### Task Summary: Configuration (Continued)

Task	Procedure
Control access to applications.	“Controlling Access to Applications” on <a href="#">page 50</a> .
Configure HTTP/HTTPS connections.	<ul style="list-style-type: none"> <li>• “Disabling an HTTP Connector” on <a href="#">page 46</a>.</li> <li>• “Enabling an HTTPS Connector” on <a href="#">page 47</a>.</li> </ul>
Configure the PureWeb® Server to use LDAP authentication.	<ul style="list-style-type: none"> <li>• “Configuring LDAP Authentication” on <a href="#">page 53</a>.</li> <li>• “Registering the LDAP Authentication Provider” on <a href="#">page 55</a>.</li> </ul>
Configure the PureWeb® Server to use database authentication.	<ul style="list-style-type: none"> <li>• “Configuring Database Authentication” on <a href="#">page 55</a>.</li> <li>• “Registering the Database Authentication Provider” on <a href="#">page 56</a>.</li> </ul>

Common cluster/node configuration tasks and procedures are listed in the table below.

### Task Summary: Cluster/Node Configuration

Task	Procedure
Inactivate a node in a cluster.	“Inactivating a Clustered/Active Node” on <a href="#">page 27</a> .
Test a node.	“Testing a Node” on <a href="#">page 27</a> .
Terminate an active rendering session.	“Terminating an Active Rendering Session” on <a href="#">page 28</a> .
Configure a cluster indirectly by modifying the .properties file.	“Configuring Clusters” on <a href="#">page 32</a> .
Configure a list of display devices by modifying the .properties file.	“Configuring Display Devices” on <a href="#">page 42</a> .
Configure applications and server interactions by modifying the .properties file.	“Configuring Application and Server Interactions” on <a href="#">page 43</a> .
Configure client and server interactions by modifying the .properties file.	“Configuring Client and Server Interactions” on <a href="#">page 44</a> .

Common application installation tasks and procedures are listed in the table below.

### Task Summary: Application Installation

Task	Procedure
Register your application by creating and adding an <application name>-plugin.xml file.	“Registering your Application” on <a href="#">page 60</a> .
Configure your application by creating and adding an <application name>-plugin.properties file.	“Configuring your Application” on <a href="#">page 66</a> .
If required, add the new resources for you application.	“Adding the Resources for your Application” on <a href="#">page 67</a> .
If new resources were added, protect the resources for your application.	Secure the resources for your application by <a href="#">“Editing the security-config.xml File to Secure Resources”</a> or by <a href="#">“Creating Private Application Resources”</a> .
Allocate the graphic processing unit (GPU) for your application.	“Allocating the Graphics Processing Unit (GPU) for your Application” on <a href="#">page 68</a> .

Common SSL certificate tasks and procedures are listed in the table below.

### Task Summary: Request/Install SSL Certificates

Task	Procedure/Step
Generate a Certificate.	Complete <a href="#">Step 1</a> of <a href="#">“Requesting and Installing an SSL Certificate”</a> .
Generate a Certificate Signing Request (CSR).	Complete <a href="#">Step 2</a> of <a href="#">“Requesting and Installing an SSL Certificate”</a> .
Submit the Certificate Signing Request to the Certificate Authority.	Complete <a href="#">Step 3</a> of <a href="#">“Requesting and Installing an SSL Certificate”</a> .
Receive the Signed Certificate from the Certificate Authority.	Complete <a href="#">Step 4</a> of <a href="#">“Requesting and Installing an SSL Certificate”</a> .
Import the Signed Certificate into a Key Store.	Complete <a href="#">Step 5</a> of <a href="#">“Requesting and Installing an SSL Certificate”</a> .
Configure the Apache Tomcat container to use the Signed Certificate from the Key Store.	“HTTP/HTTPS Connection Configuration” on <a href="#">page 46</a> .

## Chapter

# 1

## Introducing PureWeb®

This chapter contains detailed information about the PureWeb® Server, the PureWeb® solution, and its architecture. It also contains information about the system requirements for installing the PureWeb® Server.

This chapter contains the following sections:

- [About the PureWeb® Server, page 13](#)
- [Basic Architecture, page 13](#)
- [Prerequisites, page 15](#)
- [PureWeb® Server Installation, page 15](#)
- [PureWeb® Server License, page 15](#)

---

## About the PureWeb® Server

The PureWeb® Server is responsible for starting/stopping PureWeb® Services, and mediating communication between PureWeb® Clients and PureWeb® Services.

---

## Basic Architecture

The PureWeb® platform enables applications to run through a standardized web interface.

PureWeb® solutions are typically composed of three tiers:

- **PureWeb® Service**—contains most of the application logic. It uses the Service API to *plug in* to the PureWeb® Server, maintain application state, and generate rendered views.
- **PureWeb® Server**—is responsible for starting/stopping PureWeb® Services, and mediating communication between PureWeb® Clients and PureWeb® Services.

- **PureWeb® Client**—allows users to interact with PureWeb® Services through web browsers and/or mobile devices.

To PureWeb® enable an existing application means transforming the workstation version of an application into a PureWeb® Service by creating objects and calling methods from the Service API.

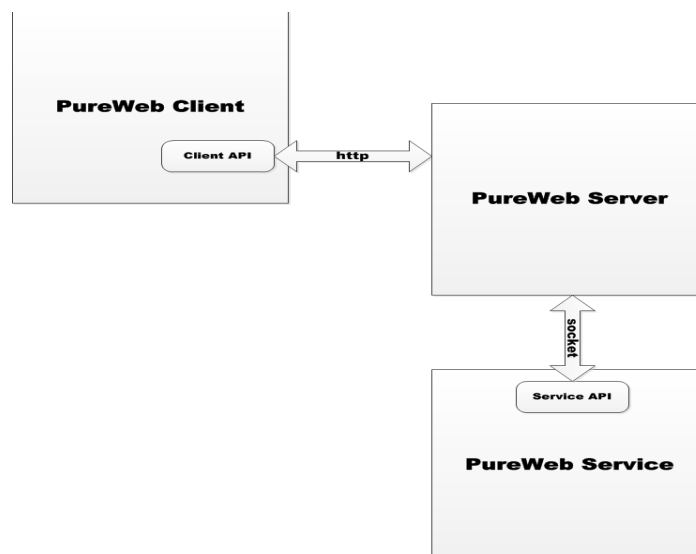
Specifically, the application will create a singleton instance of the `StateManager` class (responsible for maintaining application state in a PureWeb® Service) and pass it to a singleton instance of `StateManagerServer`.

The `StateManagerServer` handles communication with the PureWeb® Server over socket. Messages arriving from the client via the Server as XML text are converted to lists of command objects that are executed by the `StateManager`. These include commands to change the application state, commands that represent user input events, and custom commands understood by the application.

In executing the commands, the application will generate response objects including changes to the application state and new rendered images. The `StateManagerServer` converts the response objects to XML text messages that are passed back to the client via the PureWeb® Server.

Once the service creates the `StateManager` instance, it will register event handlers that are invoked when specific nodes in the application state change.

Every rendered view that is intended to be displayed on the client must be registered with the `StateManager`. In order to communicate with the `StateManager`, views must inherit from the `IRenderedView` interface. Use the adapter pattern, if it is not possible or convenient for view classes to inherit directly from the `IRenderedView` interface.



**Figure 1: Typical PureWeb® Solution**

For more information about the product, visit the PureWeb® website at [www.getpureweb.com](http://www.getpureweb.com).

---

## Prerequisites

The PureWeb® Server requires an Oracle (Sun) Java Runtime Environment (JRE) or Software Development Kit (SDK) version 1.7.

---

## PureWeb® Server Installation

When you install the PureWeb® Server on the Microsoft Windows Operating System, the default installation location is C:\CSI\PureWeb\Server however; you can change the location during installation.

---

**Warning!** The installation directory path *cannot* contain any spaces. For example, changing the default installation directory to C:\CSI\Pure Web will result in a File Not Found Exception.

---

The PureWeb® Server installation directory is referred to as PUREWEB throughout this document.

For detailed instruction on how to install PureWeb®, refer to the *PureWeb® Installation Guide for Microsoft Windows*.

---

## PureWeb® Server License

The PureWeb® Server will start, but will *not* be operational without a valid license. The license is an ASCII text file with a .lic extension and is not part of the PureWeb® Server installation. To make the PureWeb® Server operational, you must acquire the license from Calgary Scientific Inc. and follow the steps in the Installing the License procedure found in the *PureWeb® Installation Guide for Microsoft Windows*.

# 2

## Using the PureWeb<sup>®</sup> Server

This section includes information about starting, stopping, monitoring, and managing the PureWeb<sup>®</sup> Server.

This chapter contains the following sections:

- [Start the PureWeb<sup>®</sup> Server, page 16](#)
- [Stop the PureWeb<sup>®</sup> Server, page 17](#)
- [Log in to the PureWeb<sup>®</sup> Server, page 17](#)
- [Log out of the PureWeb<sup>®</sup> Server, page 18](#)
- [Monitor the PureWeb<sup>®</sup> Server, page 18](#)
- [Manage the PureWeb<sup>®</sup> Server, page 26](#)

---

## Start the PureWeb<sup>®</sup> Server

### Starting the PureWeb<sup>®</sup> Server

**Purpose:** To start the PureWeb<sup>®</sup> Server.

**Start of procedure**

1. Click on the Start PureWeb desktop icon.
- Or
1. Open a console window.
  2. Navigate to %PUREWEB\_HOME%/tomcat/bin.



3. Type startup.bat.  
After starting the server, check the  
%PUREWEB\_HOME%/tomcat/logs/pureweb.log file for errors.

End of procedure

---

## Stop the PureWeb® Server

### Stopping the PureWeb® Server

Purpose: To stop the PureWeb® Server.

Start of procedure

1. Click on the Stop PureWeb desktop icon.  
Or
1. Open a console window.
2. Navigate to %PUREWEB\_HOME%/tomcat/bin.
3. Type shutdown.bat.  
Or
1. Close the server console window.  
After stopping the server, check the system to ensure no PureWeb® processes remain.

End of procedure

---

## Log in to the PureWeb® Server

Access to the PureWeb® Server is restricted based on your security role, see Table 1 on [page 19](#). For more information about roles and access, see “Security Roles” on [page 50](#).

### Logging in to the PureWeb® Server

Purpose: To log in to the PureWeb® Server.

### Start of procedure

1. Navigate to the PureWeb® Server Login page at <http://localhost:8080/pureweb/server/login.jsp>.
2. Enter your User Name/Password.

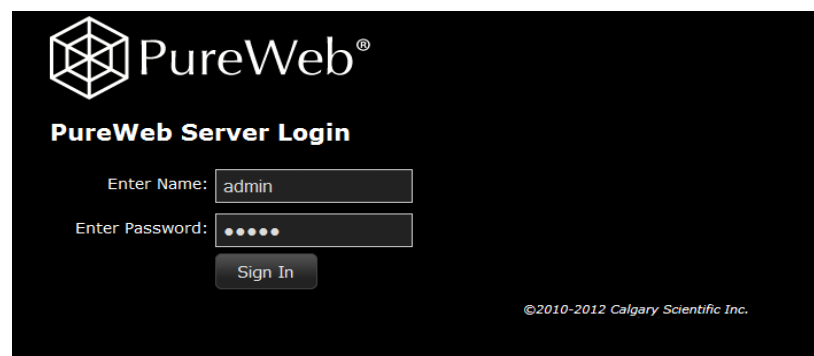
---

Note: The default User Name/Password is admin/admin.

---

3. Click the Sign In button to display the PureWeb® Server Status page, see [Figure 2](#).

### End of procedure



**Figure 2: Login**

---

## Log out of the PureWeb® Server

### Logging out of the PureWeb® Server

Purpose: To log out of the PureWeb® Server.

#### Start of procedure

1. Navigate to the PureWeb® Server Status page at <http://localhost:8080/pureweb/server/status>.
2. Click on the Logout link.  
You will be logged out of the server and the PureWeb® Server Login page will be displayed, see [Figure 2](#).

#### End of procedure

# Monitor the PureWeb® Server

You can monitor several aspects of the PureWeb® Server from the “Administrative Pages”.

## Administrative Pages

### Accessing the Server Administrative Pages

Purpose: To access the administrative pages.

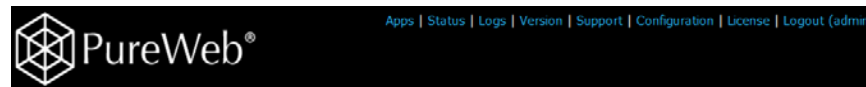
Start of procedure

1. Log in to the PureWeb® Server by following the steps in the “Logging in to the PureWeb® Server” procedure.
2. Click on the links at the top of the page, see [Figure 3](#).

Or

1. Type the Administrative URL into the address field of your browser, see [Table 1](#) on [page 19](#).

End of procedure



**Figure 3: Administrative Links**

[Table 1](#) contains a brief description of administrative pages and their URLs.

**Table 1: Administrative URLs**

URL	Description	Accessibility
<a href="http://localhost:8080/pureweb/server/login.jsp">http://localhost:8080/pureweb/server/login.jsp</a>	Displays the Login page. For more information, see “Logging in to the PureWeb® Server” on <a href="#">page 17</a> .	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Monitor</li> <li>• User</li> </ul>
<a href="http://localhost:8080/pureweb/server/links">http://localhost:8080/pureweb/server/links</a>	Displays a list of available applications. For more information see, “Accessing Application Availability Information” on <a href="#">page 23</a> .	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Monitor</li> <li>• User</li> </ul>

**Table 1: Administrative URLs (Continued)**

URL	Description	Accessibility
http://localhost:8080/pureweb/server/status	Displays a hierarchical list of all of the nodes in a cluster, all of the display devices on each node and all of the configured sessions on each display device. Active sessions also display a list of all client connections. Each active session includes a <a href="#">Release</a> link to release the session and all connections. Each active connection includes a <a href="#">Disconnect</a> link to disconnect the specific connection. Here you can make a node <a href="#">Clustered/Unclustered</a> and <a href="#">Active/Inactive</a> .	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Monitor</li> </ul>
http://localhost:8080/pureweb/server/logs	Displays a list of all available log files recorded by the PureWeb® Server and download access to each of the files. Historical logs are compressed and must be downloaded and uncompressed for viewing. For more information see, “Accessing Log Files” on <a href="#">page 25</a> .	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Monitor</li> </ul>
http://localhost:8080/pureweb/server/info	Displays build and version information about various aspects of the PureWeb® Server. For more information see, “Accessing Server Version Information” on <a href="#">page 22</a> .	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Monitor</li> <li>• User</li> </ul>
http://localhost:8080/pureweb/server/info.txt	Displays the same version information displayed in the info URL, but in raw textual format.	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Monitor</li> <li>• User</li> </ul>
http://localhost:8080/pureweb/server/support	Displays server support information. For more information, see “Accessing Support Information” on <a href="#">page 24</a> .	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Monitor</li> <li>• User</li> </ul>

**Table 1: Administrative URLs (Continued)**

URL	Description	Accessibility
http://localhost:8080/pureweb/config/plugins	Displays the links to .properties and .xml files that contain server and logging configuration information. It also contains access to system configuration information and tools to backup/restore of the system configuration. Restart the server for configuration changes to take effect. You can also change your administrative password here. For more information about the configuring the PureWeb® Server, see “Configuration Management” on <a href="#">page 29</a> .	<ul style="list-style-type: none"> <li>• Admin</li> </ul>
http://localhost:8080/pureweb/server/license	Displays license details including version, type, and expiration date. For more information see, “Accessing License Details” on <a href="#">page 24</a> .	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Monitor</li> <li>• User</li> </ul>
http://localhost:8080/pureweb/server/login.jsp?logout=true	Logs out the currently logged in user and displays the Login page. For more information, see “Logging out of the PureWeb® Server” on <a href="#">page 18</a> .	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Monitor</li> <li>• User</li> </ul>

## Accessing Cluster Status Information

Purpose: To access cluster/node information.

Start of procedure

1. Navigate to the PureWeb® Server Status page at <http://localhost:8080/pureweb/server/status>.

The page includes detailed information about clusters/nodes. For detailed information about configuring your clusters/nodes refer to [Task Summary: Cluster/Node Configuration](#), on [page 11](#).

End of procedure

## Accessing Configuration Files

Purpose: To access configuration files.

Start of procedure

1. Navigate to the PureWeb® Server Configuration page at <http://localhost:8080/pureweb/server/config/plugins>.

The page includes links to configuration files.

End of procedure

## Accessing Server Version Information

Purpose: To access information regarding your server version.

Start of procedure

1. Navigate to the PureWeb® Server Version page at <http://localhost:8080/pureweb/server/info>.

The page includes detailed information about your version of the PureWeb® Server, see [Figure 4](#).

End of procedure



Figure 4: Version Information

## Accessing Application Availability Information

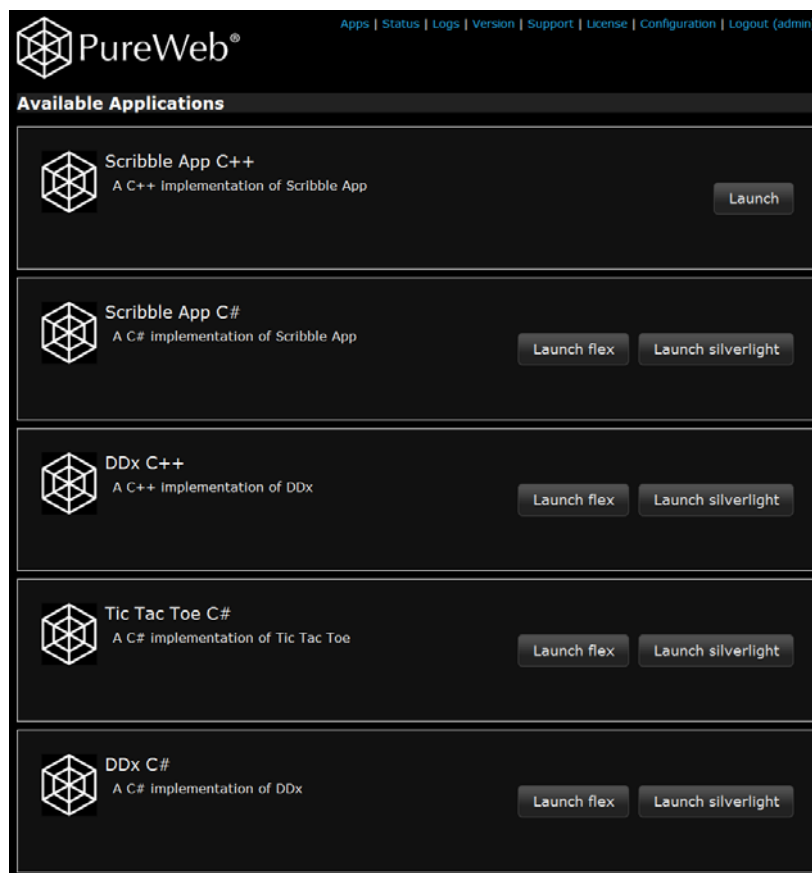
**Purpose:** To access information regarding application availability.

**Start of procedure**

1. Navigate to the PureWeb® Server App page at <http://localhost:8080/pureweb/server/links>.

The page includes detailed information about the applications that are available to you, see [Figure 5](#).

**End of procedure**



**Figure 5: Available Applications**

## Accessing Support Information

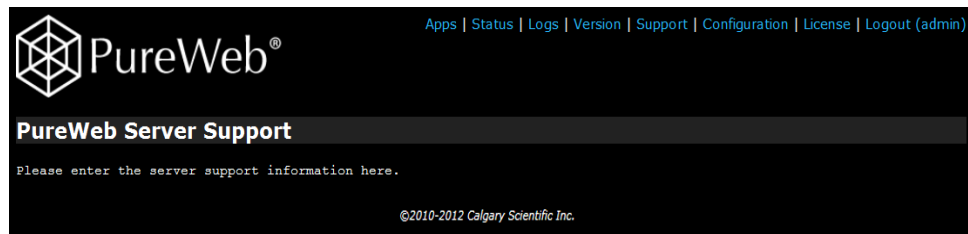
**Purpose:** To access server support information.

**Start of procedure**

1. Navigate to the PureWeb® Server Support page at <http://localhost:8080/pureweb/server/support>.

The page includes detailed contact information for PureWeb® Server support, see [Figure 6](#).

**End of procedure**



**Figure 6: Support Information**

## Accessing License Details

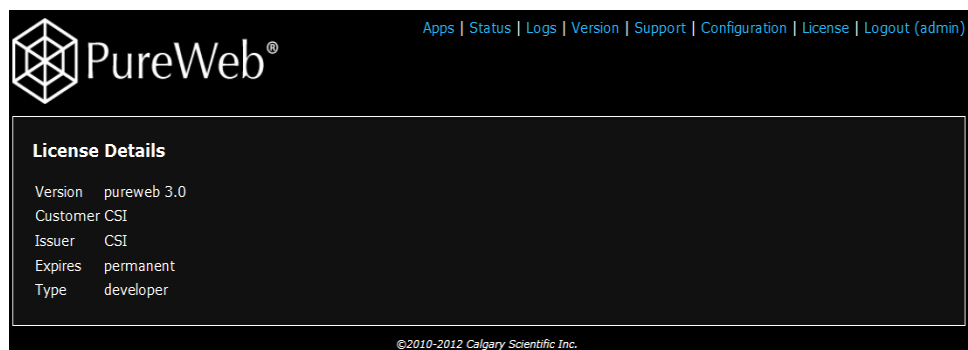
**Purpose:** To access your server license details.

**Start of procedure**

1. Navigate to the PureWeb® Server License page at <http://localhost:8080/pureweb/server/license>.

The page includes detailed information about your PureWeb® Server license, see [Figure 7](#).

**End of procedure**



**Figure 7: License Details**



## Accessing Log Files

**Purpose:** To access log files.

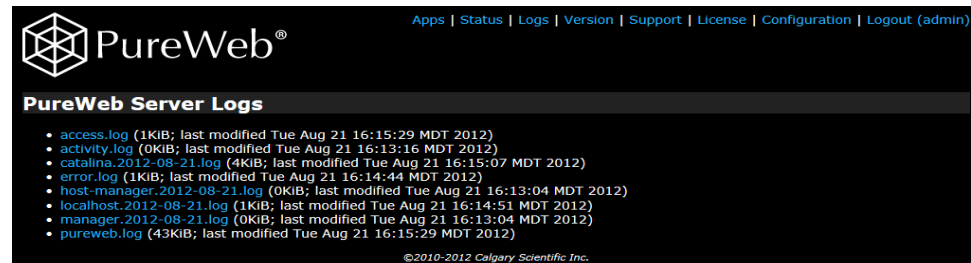
### Start of procedure

1. Navigate to the PureWeb® Server Logs page at <http://localhost:8080/pureweb/server/logs>.

The page includes links to current and archived log files, see Figure 8 on [page 25](#). For detailed information about the location and contents of each log file, refer to [Table 2](#). To modify a log file retention period follow the steps in the “[Configuring Log File Retention Period](#)” procedure.

2. Click on the link to view the current logs directly.
3. Download the compressed archived logs for offline examination.

### End of procedure



**Figure 8: Logs**

[Table 2](#) contains a list of the location of each log file and its contents.

**Table 2: System Log Files**

Location	Description
%PUREWEB_HOME%/tomcat/logs/pureweb.log	This file contains all of the messages from the PureWeb® Server about startup, shutdown, configuration details, and client activities. Messages written to this log file indicate their severity as ERROR, WARN, INFO or DEBUG from highest to lowest severity.
%PUREWEB_HOME%/tomcat/logs/error.log	This file provides a summary of any errors and warnings logged by the PureWeb® Server. Messages logged to this file also appear in the pureweb.log file.
%PUREWEB_HOME%/tomcat/logs/access.log	This file provides PureWeb® Server access information. Messages logged to this file also appear in the pureweb.log file.

**Table 2: System Log Files (Continued)**

Location	Description
%PUREWEB_HOME%/tomcat/logs/activity.log	This file provides a summary of application activity logged by the PureWeb® Server and cooperating applications. This includes application startup and shutdown and additional events deemed to be significant by the specific application. Messages logged to this file also appear in the pureweb.log file.
%PUREWEB_HOME%/tomcat/logs/catalina.log	This file contains messages from the Java Virtual Machine in which the Apache Tomcat container and PureWeb® Server run.
%PUREWEB_HOME%/tomcat/logs/manager.log	This file is generated by Apache Tomcat.
%PUREWEB_HOME%/tomcat/logs/host-manager.log	This file is generated by Apache Tomcat.
%PUREWEB_HOME%/tomcat/logs/localhost.log	This file contains information from the localhost Apache Tomcat engine.

---

## Manage the PureWeb® Server

Manage several aspects of the PureWeb® Server from the Status page including:

- “Cluster Management” on [page 26](#)
- “Configuration Management” on [page 29](#)

### Cluster Management

The Status page includes information used to manage clusters and nodes, see [Figure 9](#). The [Clustered/Unclustered](#) and [Active/Inactive](#) states can be used to allow existing sessions to complete gracefully before shutting down a node for maintenance, or for testing a new node before bringing it into service. For detailed information about clusters, see “Cluster Plugins” on [page 31](#).

For more information, see the procedures below.

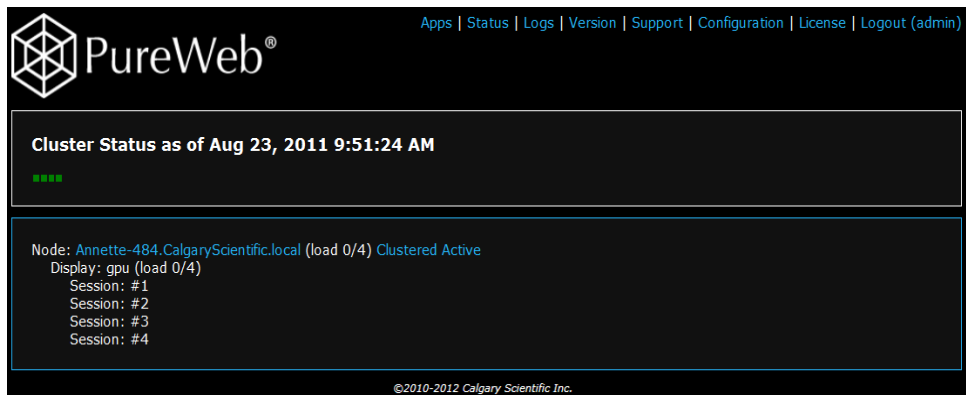


Figure 9: Cluster Status

## Inactivating a Clustered/Active Node

**Purpose:** To gracefully shut down a clustered/active node. Note that changes made here are not persistent. When you restart the server, the values saved in the cluster-plugin.xml file are used.

### Start of procedure

1. Navigate to the PureWeb® Server Status page available at <http://localhost:8080/pureweb/server/status>.
2. Click the [Inactive](#) link associated with the [Clustered/Active](#) node that you want to inactivate, see [Table 3](#) for details.  
This will redirect any requests for new sessions to another [Clustered/Active](#) node.
3. Once all existing sessions have completed, mark the node as [Unclustered](#) so that it will not receive any further requests.
4. Shut down the node.

### End of procedure

## Testing a Node

**Purpose:** To test a node before bringing it into service.

### Start of procedure

1. Navigate to the PureWeb® Server Status page available at <http://localhost:8080/pureweb/server/status>.
2. Configure the test node as [Unclustered/Active](#), see [Table 3](#).
3. Start the node for testing.

4. Once testing is complete, shut down the node.
5. Reconfigure the node as [Clustered/Active](#).
6. Restart the node to bring it into full service.

End of procedure

**Table 3: Cluster Status**

Status	Description
Clustered	Nodes that are participating in the cluster include a <a href="#">Clustered</a> link used to put the node into an <a href="#">Unclustered</a> state. <a href="#">Clustered</a> nodes redirect requests for new sessions to the least busy node in the cluster.
Unclustered	Nodes that are <i>not</i> participating in the cluster include an <a href="#">Unclustered</a> link used to put the node into a <a href="#">Clustered</a> state. <a href="#">Unclustered</a> nodes <i>do not</i> redirect requests for new sessions to other nodes in the cluster.
Active	Nodes that accept requests for new sessions include an <a href="#">Active</a> link used to put the node into an <a href="#">Inactive</a> state. <a href="#">Active</a> nodes accept requests for new sessions if they are among the least busy nodes in the cluster.
Inactive	Nodes that <i>do not</i> accept requests for new sessions will include an <a href="#">Inactive</a> link used to put the node into an <a href="#">Active</a> state. <a href="#">Inactive</a> nodes <i>do not</i> accept requests for new sessions.

## Terminating an Active Rendering Session

Purpose: To terminate an active rendering session.

Start of procedure

1. Navigate to the PureWeb® Server Status page available at <http://localhost:8080/pureweb/server/status>.
2. Click the [Release](#) and [Disconnect](#) links associated with the active session that you want to terminate, see [Table 4](#) for details.

End of procedure

**Table 4: Active Rendering Session**

Name	Description
Release	Each active session includes a <a href="#">Release</a> link used to forcibly release the session. This disconnects all connected clients and shutdowns the associated process.
Disconnect	Each client connection includes a <a href="#">Disconnect</a> link used to forcibly disconnect the specific client. The session is automatically released when the last client is disconnected.

## Configuration Management

Use the links to .properties and .xml files on the Configuration page (see [Figure 10](#)) to configure the following:

- “[Server Configuration](#)”
- “[Logging Configuration](#)”
- “[System Configuration](#)”

**Figure 10: Configuration**

## Server Configuration

The Server Configuration section of the Configuration page allows you to configure both “[Application Plugins](#)” and “[Cluster Plugins](#)”.

### Application Plugins

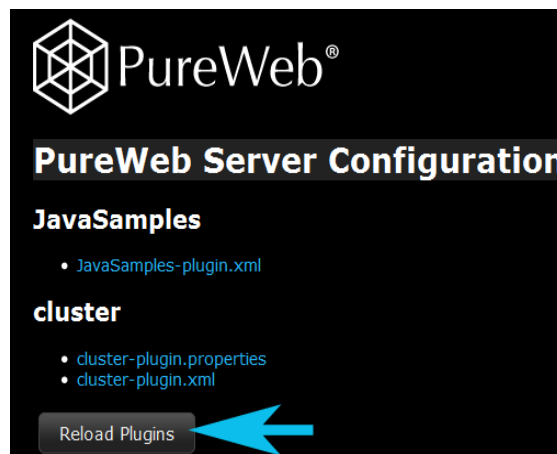
A plugin is a set of software components that once installed adds a narrow set of abilities to a larger application. Each installed plugin has links to its configuration properties and xml definitions. Generally, you will make changes to the configuration properties however, in some situations the plugin definitions may also need to be changed. Customize application plugins by following the steps in the “[Configuring an Application Plugin](#)” procedure.

### Reloading a Plugin

**Purpose:** When you are running or debugging the PureWeb® STK sample applications that we have provided, you may find it necessary to reload a plugin. To reload a plugin follow the steps provided below.

#### Start of procedure

1. Open a browser.
2. Navigate to <http://localhost:8080/pureweb/config/plugins>.
3. Enter admin/admin into the Name and Password fields.
4. Click the Sign In button to open the Configuration page.
5. Click the Reload Plugins button.



**Figure 11: Reload Plugins**

---

Note: The configuration files displayed in the image above may not be identical to the list displayed on your system. The list is comprised of configuration files unique to your system.

---

End of procedure

## Configuring an Application Plugin

Purpose: To configure a plugin.

Start of procedure

1. Navigate to the PureWeb® Server Configuration page at <http://localhost:8080/pureweb/config/plugins>.
2. Click the link to the configuration properties or xml definitions of the plugin that you want to modify.  
This opens a simple text editor with Save and Cancel buttons.
3. Click Save after making modifications to the file.
4. Click Reload Plugins for the changes to take effect.  
Any plugins that have pending configuration changes will display reload required indicating that changes have *not* been applied.

End of procedure

### Cluster Plugins

The %PUREWEB\_HOME%/conf/cluster-plugin.xml file defines the clustered application hosting facilities provided by the PureWeb® Server. Follow the steps in the “[Configuring Clusters](#)” procedure to set values in the %PUREWEB\_HOME%/conf/cluster-plugin.properties file to configure this component.

## Configuring Clusters

**Purpose:** To configure several PureWeb® Servers together into a cluster by modifying the cluster-plugin.properties file.

### Start of procedure

1. Navigate to the PureWeb® Server Configuration page at <http://localhost:8080/pureweb/config/plugins>.
2. Click the cluster-plugin.properties link.  
This opens a simple text editor with Save and Cancel buttons.
3. Use the information provided in [Table 5](#) to modify the file.
4. Click Save.
5. Click Reload Plugins for the changes to take effect.  
Any plugins that have pending configuration changes will display reload required indicating that changes have *not* been applied.

### End of procedure

Use the properties in [Table 5](#) to group several PureWeb® Servers together into a cluster defined by a specific cluster address and port.

**Table 5: Cluster Properties**

Name	Description
cluster.enabled	Controls whether PureWeb® clustering is enabled or disabled. All enabled nodes with the same cluster address and port will form a cluster and balance client load across the cluster.
cluster.interface	Defines the multicast interface that will be bound to for all multicast communications. Leaving blank works, but with multi-NIC systems the bound interface will be indeterminate.
cluster.address	Defines the multicast address shared by all nodes in a cluster. Valid addresses must be in the range 224.0.0.0 through 239.255.255.255. Nodes will only form a cluster if they have the same cluster address and port.
cluster.port	Defines the multicast port shared by all nodes in a cluster. Valid ports must be in the range 0 through 65535 and should avoid the privileged ports with values below 1024. Nodes will only form a cluster if they have the same cluster address and port.



**Table 5: Cluster Properties (Continued)**

Name	Description
cluster.status.broadcast	Defines the number of seconds between status message broadcasts to the cluster. All nodes in a cluster send out periodic status updates notifying the other cluster nodes of their existence and current load conditions. These updates are broadcast whenever status changes occur on a node in addition to the frequency defined by this value. The standard value for this timeout is 25 seconds.
cluster.status.timeout	Defines the number of seconds that a status message from another node in cluster is to be considered valid. The source node is marked as unresponsive if no status update is received by this timeout. This value should be set slightly higher than the status broadcast timeout to ensure that status updates are received and processed before a node is marked as unresponsive. The standard value for this timeout is 30 seconds.
cluster.response.timeout	<p>Define the length of time that the middle tier should wait for responses from the service. Note that this property is used even if clustering is disabled.</p> <p>We recommend that you keep this value small, but never less than three seconds. When debugging increase the value to prevent the middle tier from closing the communication channels on a processed that has been stepped into with a debugger.</p> <p><b>Example:</b> cluster.response.timeout=10</p>
application.connection.address	Defines the address for PureWeb 3 Socket based service connections.
application.connection.port	Defines the port for PureWeb 3 Socket based service connections.

---

Note: New connections may be redirected to cluster nodes from which status updates have not been received within this timeout. The node selection algorithm should exclude these nodes.

---

End of procedure

## Cluster Properties

All nodes that are configured with the same cluster.address and cluster.port cooperate within a cluster and redirect requests as required to ensure that the load is distributed evenly.

**Table 6: Cluster Properties**

Name	Description
cleanupTimeout	Defines the amount of time to wait between periodic checks for inactive or abandoned client sessions. The default is 5 seconds.

See Table 5 on [page 32](#) for descriptions of other clustering properties.

## Node Properties

The PureWeb® Server will attempt to determine its hostname automatically, but occasionally there are system or network configuration problems that prevent this from being possible. In these cases, configure the hostname explicitly.

**Table 7: Node Properties**

Name	Description
hostname	Defines the hostname of this cluster node. This value is optional and will default to the system's hostname. The specified hostname must be resolvable via either the local hosts file or the configured DNS system.
clustered	Defines whether this node is participating in the cluster defined by the cluster address and port properties. Clustered nodes may redirect requests to other nodes to ensure load is evenly distributed across the cluster. Unclustered nodes will not redirect requests to other nodes and may report Service Unavailable errors if they have no available sessions.
active	Defines whether this node is active and will accept new connections or not. Active nodes will accept new connections if they have available sessions. Inactive nodes will not accept new connections even if they have available sessions.

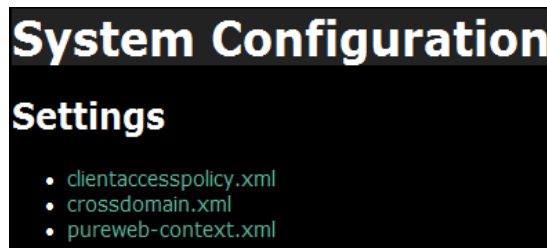
See Table 9 on [page 43](#) for descriptions of other node properties.

## Making a Service Available Across Domains

**Purpose:** To control the domain access of clients. This procedure is recommended for those using a cluster configuration.

## Start of procedure

1. Navigate to the server Configuration page:
  - Non-SSL:  
http://<hostname>:8080/pureweb/config/plugins
  - SSL:  
https://<hostname>:8443/pureweb/config/plugins
2. Click on the clientaccesspolicy.xml file link to open the file.



**Figure 12: Cross Domain Configuration**

3. Edit the file by replacing lines 06 and 07 with the following.
 

```
<domain uri="http://*.DOMAIN.NAME" />
<domain uri="https://*.DOMAIN.NAME" />
```

### Example:

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <access-policy>
03   <cross-domain-access>
04     <policy>
05       <allow-from http-request-headers="*" http-methods="*">
06         <domain uri="http://*" />
07         <domain uri="https://*" />
08       </allow-from>
09     <grant-to>
10       <resource path="/" include-subpaths="true"/>
11     </grant-to>
12   </policy>
13 </cross-domain-access>
14 </access-policy>
```

4. Click Save.
5. Click on the crossdomain.xml file link to open the file.
6. Edit the file by replacing lines 05 with the following.
 

```
<allow-access-from domain="*.DOMAIN.NAME"/>
```

### Example:

```
01 <?xml version="1.0"?>
02 <!DOCTYPE cross-domain-policy SYSTEM
    "http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
03 <cross-domain-policy>
04   <site-control permitted-cross-domain-policies="master-only"/>
```

```
05 <allow-access-from domain="*" />
06 </cross-domain-policy>
```

7. Click Save.

End of procedure

## Logging Configuration

Access the current and archived log files by following the steps in the “[Accessing Log Files](#)” procedure. The logging configuration defines several different log files described in Table 2 on [page 25](#). Each file is configured to be compressed and archived daily so that the active log file will only contain the current day's information. The default retention policy in this configuration is to keep the most recent 14 days of log files. Production installations that require different retention policies can change this configuration to suit their requirements, by following the steps in the “[Configuring Log File Retention Period](#)” procedure.

## Configuring Log File Retention Period

**Purpose:** To change the log file retention period by modifying the %PUREWEB\_HOME%/tomcat/custom/classes/logback.properties file.

Start of procedure

1. Navigate to the PureWeb® Server Configuration page at <http://localhost:8080/pureweb/config/plugins>.
2. Click the logback.properties link to open the Logging Configuration page.



**Figure 13: Logging Configuration**

3. Enter the numbers of retention days for each log file.
4. Click Save to commit the changes.

5. Click Reload Logging for the changes to take effect.

Any pending configuration changes will display reload required indicating that changes have *not* been applied.

End of procedure

## System Configuration

The system configuration section of the PureWeb® Server Configuration page provides the following functionality:

- [“Changing the Administrator Password”](#)
- [“Backing Up the PureWeb® Server”](#)
- [“Restoring the PureWeb® Server from Backup”](#)
- [“Changing Server Support Information”](#)

## Changing the Administrator Password

Purpose: To change the administrator password.

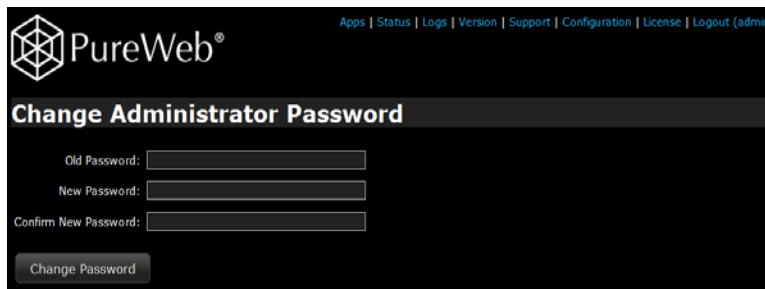
Start of procedure

1. Navigate to the PureWeb® Server Configuration page at <http://localhost:8080/pureweb/config/plugins>.
2. Click the Change Administrator Password link.
3. Enter your old and new password information.

The correct old password must be entered and the new password must match the confirmed new password. New passwords must be at least 9 characters long, see [Figure 14](#).

4. Click Change Password.
5. Return to the PureWeb® Server Configuration page.
6. Click ReloadPureWeb System for the changes to take effect.

End of procedure



**Figure 14: Change Administrator Password**

## Backing Up the PureWeb® Server

**Purpose:** To backup the exist PureWeb® Server configuration.

**Start of procedure**

1. Navigate to the PureWeb® Server Configuration page at <http://localhost:8080/pureweb/config/plugins>.
2. Click the Backup/Restore Configuration link.
3. Click Save New Backup, see [Figure 15](#).

**End of procedure**

## Restoring the PureWeb® Server from Backup

**Purpose:** To restore the PureWeb® Server to a previous configuration.

**Prerequisites**

- You must have created a backup configuration using the steps in the “[Backing Up the PureWeb® Server](#)” procedure.

**Start of procedure**

1. Navigate to the PureWeb® Server Configuration page at <http://localhost:8080/pureweb/config/plugins>.
2. Click the Backup/Restore Configuration link.
3. Select an existing backup to overwrite the current configuration, see [Figure 15](#).
4. Restart the PureWeb® Server for the changes to take effect.

**End of procedure**

From the Configuration Backups page, you can also delete existing backups permanently or export them to the administrator's client machine and save them there. You can import backups from the administrator's client machine and restored, to establish common configuration on one PureWeb® Server, then exported and imported to several other PureWeb® Servers simplifying the task of configuring a multi-node PureWeb® cluster.

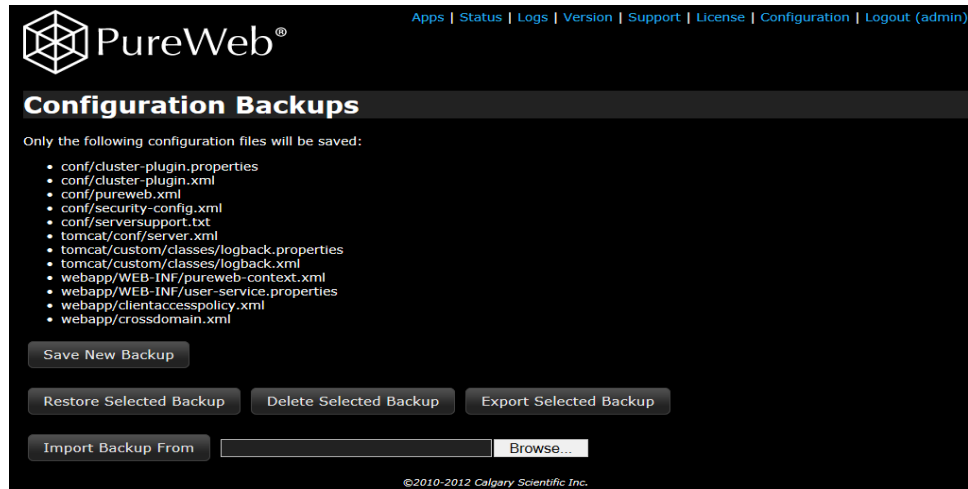


Figure 15: Backup/Restore Configuration

## Changing Server Support Information

**Purpose:** To customize your PureWeb® Server Support page information.

**Start of procedure**

1. Navigate to the PureWeb® Server Configuration page at <http://localhost:8080/pureweb/config/plugins>.
2. Click the Change Server Support Information link.  
This opens a simple text editor with Save and Cancel buttons.
3. Edit the file to include your specific support information.
4. Click Save to commit the changes.

**End of procedure**

# 3

## Configuring the PureWeb<sup>®</sup> Server

You configure the PureWeb<sup>®</sup> Server using several different configuration files offering various levels of simplicity, flexibility and detail. Restart the server to reload changes to any of the configuration files described below or dynamically reload changes to individual configuration files using the configuration management facilities.

This chapter contains the following sections:

- [Configuration Hierarchy, page 41](#)
- [Configuration Properties, page 42](#)
- [HTTP/HTTPS Connection Configuration, page 46](#)
- [Security Configuration, page 48](#)
- [LDAP Authentication Configuration, page 53](#)
- [Database Authentication Configuration, page 55](#)
- [Reverse Proxies and Load Balancers, page 57](#)

---

**Warning!** When using Windows Server 2008, you must modify the User Access Control (UAC) settings, so that the user can edit and save configuration files. If the user has the wrong permissions, and they attempt to save a modified configuration file, an Internal Server Error is displayed.

---



## Configuration Hierarchy

The PureWeb® Server is implemented as a standard Java Web Application, using common Java components including the Apache Tomcat 7.0 web container (<http://tomcat.apache.org>) and various libraries from the Spring Framework (<http://www.springsource.org>).

Configuration of the PureWeb® Server is organized into a hierarchy of configuration files defined by these technologies and the underlying plugin architecture defined by the PureWeb® Server. Installation of optional plugins will add plugin specific .xml and .properties files to the list of configuration files in %PUREWEB\_HOME%/conf. Table 8 briefly describes the content of each of the configuration files.

**Table 8: Configuration Hierarchy**

Configuration File	Description
%PUREWEB_HOME%/webapp/WEB-INF/web.xml	Standard Java Servlet Specification 2.4 Web Application definition loaded by Apache Tomcat. Defines the PureWeb® Server web application.
%PUREWEB_HOME%/webapp/WEB-INF/pureweb-context.xml	Standard Spring Framework Root Web Application Context loaded by the web application defined in web.xml. The PureWeb® Context defines the authentication facilities used by all components within the PureWeb® Server.
%PUREWEB_HOME%/webapp/WEB-INF/dispatcher-servlet.xml	Standard Spring Framework Servlet Dispatcher loaded by the web application defined in web.xml. Defines common facilities used by various components of the PureWeb® Server.
%PUREWEB_HOME%/conf/security-config.xml	Additional security configuration allowing various plugins to define their own security requirements using Java annotations.
%PUREWEB_HOME%/conf/pureweb.xml	Common logging, error handling and configuration management facilities provided by the PureWeb® Server.
%PUREWEB_HOME%/conf/cluster-plugin.xml	Clustered application hosting facilities.
%PUREWEB_HOME%/conf/cluster-plugin.properties	Basic configuration properties referenced by the cluster-plugin.xml file above.

---

# Configuration Properties

The %PUREWEB\_HOME%/conf/cluster-plugin.properties file defines basic application hosting and clustering configuration properties. These properties are used by the lower level “Cluster Properties” on [page 33](#) configuration file to control specific features provided by the PureWeb® Server.

## Configuring Display Devices

**Purpose:** To configure the display devices available to the server.

### Start of procedure

1. Navigate to the PureWeb® Server Configuration page at <http://localhost:8080/pureweb/config/plugins>.

2. Click the cluster-plugin.properties link.

This opens a simple text editor with Save and Cancel buttons.

3. Use the information provided in [Table 9](#) to modify the file.
4. Click Save.
5. Click Reload Plugins for the changes to take effect.

Any plugins that have pending configuration changes will display reload required indicating that changes have *not* been applied.

### End of procedure

Use the properties in [Table 9](#) to create a list of display devices available to the server.

**Table 9: Display Properties**

Name	Description	Format	Windows Example
display.list	Defines the list of display devices available for use by this server. Each entry specifies the maximum number of client sessions allowed on a specific display device.	<p>&lt;max_sessions&gt;@&lt;display_device&gt;</p> <p>&lt;max_sessions&gt; is a positive integer (1..N)</p> <p>&lt;display_device&gt; is the name of the associated display device.</p> <p>Entries for multiple display devices must be separated by spaces.</p>	<p>Display token form = gpu or gpu.n</p> <p>n is the GPU instance when using Quadro or ATI GPUs.</p> <p>Example: display.list=4@gpu.0 4@gpu.1</p> <p>Permits 4 users to share the first Quadro or ATI GPU and 4 users to share the second Quadro or ATI GPU.</p> <p>Example: display.list=4@gpu</p> <p>Permits 4 users to share the available GPU resources on the system when a Quadro or ATI GPU is not being used</p>

## Configuring Application and Server Interactions

**Purpose:** To configure application and server interactions.

### Start of procedure

1. Navigate to the PureWeb® Server Configuration page at <http://localhost:8080/pureweb/config/plugins>.
2. Click the cluster-plugin.properties link.  
This opens a simple text editor with Save and Cancel buttons.
3. Use the information provided in [Table 10](#) to modify the file.
4. Click Save.
5. Click Reload Plugins for the changes to take effect.  
Any plugins that have pending configuration changes will display reload required indicating that changes have *not* been applied.

### End of procedure

Use the properties in [Table 10](#) to create rules for application and server interactions.

**Table 10: Process Properties**

Name	Description	Default Values
process.response.timeout	Defines the maximum number of seconds that an application process has to respond to a request from the server. If the application fails to respond within this timeout period, the server will consider it to be unresponsive and kill it.	30 Seconds
process.shutdown.timeout	Defines the maximum number of seconds that an application process has to cleanly shutdown when requested by the server. If the application fails to shutdown within this timeout period, the server will consider it to be unresponsive and kill it.	30 Seconds
process.cleanup.timeout	Defines the number of seconds between checks for inactive processes. Inactive processes will be released back to the cluster by the cleanup process.	5 Seconds

## Configuring Client and Server Interactions

Purpose: To configure client and server interactions.

Start of procedure

1. Navigate to the PureWeb® Server Configuration page at <http://localhost:8080/pureweb/config/plugins>.
2. Click the cluster-plugin.properties link.  
This opens a simple text editor with Save and Cancel buttons.
3. Use the information provided in [Table 11](#) to modify the file.
4. Click Save.

5. Click Reload Plugins for the changes to take effect.

Any plugins that have pending configuration changes will display reload required indicating that changes have *not* been applied.

End of procedure

Use the properties in [Table 11](#) to create rules for client and server interactions.

**Table 11: Client Properties**

Name	Description	Default Values
client.activity.timeout	Defines the maximum number of seconds that a client can consume a session with no client activity. If there is no activity from a client within this timeout the application process will be terminated and the session released to be available to other users.	30 Seconds
user.process.limit	Defines the maximum number of rendering processes that each user of the system may have. Requests for rendering processes above this limit will be rejected. A value of 0 may be used to specify no process limit.	1
silverlight.socket.policy.enabled	Set to true to enable the Microsoft Silverlight policy server or false to disable it.	enabled
silverlight.socket.policy.address	Defines the address that the xmlsocket policy server will listen on for permitting Microsoft Silverlight clients to connect using a raw (high-performance) socket.	
silverlight.socket.policy.port	Defines the port that the xmlsocket policy server will listen on for permitting Microsoft Silverlight clients to connect using a raw (high-performance) socket. The Windows default is 943. This will require port redirection from 943->8943 (see tools/iptables.sh).	
silverlight.socket.policy.hosts	Defines the domain and port patterns that Microsoft Silverlight clients will be permitted to connect to using a raw (high-performance) socket. Connections are only permitted on ports in the 4502 - 4534 range.	*:4502 (all domains on port 4502)

---

# HTTP/HTTPS Connection Configuration

You configure the Apache Tomcat container used by the PureWeb® Server in the %PUREWEB\_HOME%/tomcat/conf/server.xml file. This file contains details of the HTTP connectors used by the PureWeb® Server to receive connections from PureWeb® Clients.

By default, there are two HTTP connectors are enabled. One connector listens on the standard Apache Tomcat port 8080 and another listens on port 4502. This is specifically for those Microsoft Silverlight clients that use raw socket connections. Microsoft Silverlight clients that do not use raw socket connections can use the port 8080 connector. You can then disable the port 4502 connector by following the steps in the [“Disabling an HTTP Connector”](#) procedure.

## Disabling an HTTP Connector

**Purpose:** To disable the port 4502 connector for Microsoft Silverlight clients that do not use raw socket connections.

### Start of procedure

1. Navigate to the %PUREWEB\_HOME%/tomcat/conf/server.xml file.
2. Comment out lines 109-115 in the code shown below.
3. Save the file before closing.

### Example:

```
100: <!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
101: <Connector port="8080" address="0.0.0.0"
102:  maxHttpHeaderSize="8192" emptySessionPath="false"
103:  maxKeepAliveRequests="-1"
104:  maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
105:  enableLookups="false" redirectPort="8443" acceptCount="100"
106:  connectionTimeout="600000" disableUploadTimeout="true"
107:  useBodyEncodingForURI="true" URIEncoding="UTF-8" />
108: <!-- Define a non-SSL HTTP/1.1 Connector on port 4502 for Microsoft Silverlight socket connections -->
109: <Connector port="4502" address="0.0.0.0"
110:  maxHttpHeaderSize="8192" emptySessionPath="false"
111:  maxKeepAliveRequests="-1"
112:  maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
113:  enableLookups="false" redirectPort="8443" acceptCount="100"
114:  connectionTimeout="600000" disableUploadTimeout="true"
115:  useBodyEncodingForURI="true" URIEncoding="UTF-8" />
```

4. Restart the server for the changes to take effect.

### End of procedure

## Enabling an HTTPS Connector

**Purpose:** To enable an HTTPS connector in the Apache Tomcat container.

### Prerequisites

- A server certificate is required. For more information about server certificates, see Appendix A, “Secure Socket Layer (SSL) Certificate,” on [page 72](#).

### Start of procedure

1. Navigate to the %PUREWEB\_HOME%/tomcat/conf/server.xml file.
2. Remove the comments on lines 133 and 151 in the “[SSL Example](#)” shown below.
3. Ensure that the keystoreFile, keystorePass, and keystoreType values on line 09 are set appropriately.

---

**Note:** If you created your certificate using the “[Requesting and Installing an SSL Certificate](#)” procedure, you must modify the keystoreType value on line 141.

---

4. Optionally, restrict the set of available ciphers to those that are cryptographically strong by including the ciphers configuration on lines 153-159.
5. Save the file before closing.

---

**Note:** The default port for SSL configuration is port 8443. If you have port 8443 blocked, configure port 80 instead by replacing line 134 in the example below as follows:

```
<Connector port="80" address="0.0.0.0"
```

Use port 80 when “[Accessing the Server Administrative Pages](#)”. For example: <https://localhost:80/pureweb/server/login.jsp>.

---

### SSL Example

```
132: <!-- Define a SSL HTTP/1.1 Connector on port 8443 -->
133: <!--
134: <Connector port="8443" address="0.0.0.0" SSLEnable="true"
135:   maxHttpHeaderSize="8192" emptySessionPath=="false"
136:   maxKeepAliveRequests="-1"
137:   maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
138:   enableLookups="false" disableUploadTimeout="true"
139:   acceptCount="100" scheme="https" secure="true"
140:   clientAuth="false" sslProtocol="TLS"
```

```
141: keystoreFile="conf/cert.p12" keystorePass="password" keystoreType="PKCS12"
142: useBodyEncodingForURI="true" URIEncoding="UTF-8"
143: ciphers="SSL_RSA_WITH_RC4_128_MD5,
144:     SSL_RSA_WITH_RC4_128_SHA,
145:     TLS_RSA_WITH_AES_128_CBC_SHA,
146:     TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
147:     TLS_DHE_DSS_WITH_AES_128_CBC_SHA,
148:     SSL_RSA_WITH_3DES_EDE_CBC_SHA,
149:     SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA,
150:     SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA"/>
151: -->
152: <!-- enable specific (non-weak) ciphers -->
153: <!-- ciphers="SSL_RSA_WITH_RC4_128_MD5, SSL_RSA_WITH_RC4_128_SHA,
154:     TLS_RSA_WITH_AES_128_CBC_SHA,
155:     TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
156:     TLS_DHE_DSS_WITH_AES_128_CBC_SHA,
157:     SSL_RSA_WITH_3DES_EDE_CBC_SHA,
158:     SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA,
159:     SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA" -->
```

End of procedure

---

## Security Configuration

The PureWeb® Server is configured to restrict access to pages intended for server monitoring and administration. Requests for access to such pages will require user authentication and will only be allowed to authorized administrative users. These security policies are configured in the %PUREWEB\_HOME%/webapp/WEB-INF/pureweb-context.xml file.

### Authentication Options

[Table 12](#) describes the two different authentication options that are available to configure the type of challenge that the PureWeb® Server presents to unauthenticated clients.



**Table 12: Authentication Options**

Name	Description
Basic Authentication	Using basic authentication the PureWeb® Server will report an HTTP 401 Unauthorized error to unauthenticated clients. Standard web browsers will display an authentication dialog requesting the username and password.
Form-Based Authentication	Using form-based authentication the PureWeb® Server will redirect unauthenticated clients to an authentication page to request the username and password.

Form-based authentication is recommended and is selected by default, but may be changed in the pureweb-context.xml file.

## Configuring the Authentication Type

**Purpose:** To change from form-based to basic authentication. We recommend that you use basic authentication only if you are also using SSL.

### Start of procedure

1. Navigate to the %PUREWEB\_HOME%/webapp/WEB-INF/pureweb-context.xml file.
2. In the code below, comment out the form-based authentication section in lines 37 to 42.
3. Remove the comments on lines 44 and 47.

### Example:

```

32: <s:http auto-config="true" realm="PureWeb" createSession="always">
33: <s:request-cache ref="requestCache"/>
34: <!-- do not copy session attributes on authentication (in particular Client instances) -->
35: <s:session-management session-fixation-protection="newSession"/>
36: <!-- form-based authentication -->
37: <s:form-login login-processing-url="/login"
38:   login-page="/pureweb/server/login.jsp"
39:   authentication-failure-ref="authenticationFailureHandler"
40:   default-target-url="/pureweb/server/status"
41:   always-use-default-target="false"/>
42: <s:logout logout-url="/logout" logout-success-url="/pureweb/server/logout.jsp"/>
43: <!-- basic authentication -->
44: <!--
45: <s:http-basic/>
46: <s:logout logout-url="/logout"/>

```

47: --&gt;

End of procedure

## Security Roles

The PureWeb® Server defines four primary security roles used for controlling access to different features. [Table 13](#) describes these four security roles.

**Table 13: Security Roles**

Name	Description
PureWeb® Server Admin	This role is required to access various administrative features of the PureWeb® Server such as viewing status information and downloading log files. In the various configuration files, this role is defined as <code>ROLE_PUREWEB_SERVER_ADMIN</code> .
PureWeb® Server Monitor	This role has reduced access to various administrative features of the PureWeb® Server such as viewing status information and downloading log files. In particular, this user is unable to access server configuration pages. In the various configuration files, this role is defined as <code>ROLE_PUREWEB_SERVER_MONITOR</code> .
PureWeb® User	This role is required to access application specific features of the PureWeb® Server such as running applications and viewing information. In the various configuration files, this role is defined as <code>ROLE_PUREWEB_USER</code> .
PureWeb® Collaborator	This role is required to access the collaboration features of the PureWeb® Server. In the various configuration files, this role is defined as <code>ROLE_PUREWEB_COLLABORATOR</code> .
PureWeb® User Admin	This role is required to access application specific features of the WebReport client application. In the various configuration files, this role is defined as <code>ROLE_PUREWEB_USER_ADMIN</code> .

## Controlling Access to Applications

**Purpose:** To control access to PureWeb® applications by editing the `pureweb-context.xml` file.

**Start of procedure**

1. Navigate to the `%PUREWEB_HOME%/webapp/WEB-INF/pureweb-context.xml` file.
2. Edit the values of the `intercept-url` pattern and the access elements.

**Example:**

```
<s:intercept-url pattern="/pureweb/app/**" access="hasRole('ROLE_PUREWEB_USER')"/>
```

3. Save the changes before closing the file.

End of procedure

**Adding PureWeb Users**

**Purpose:** To add users by editing the user-service.properties file.

Start of procedure

1. Generate a new password by following the steps in the [“Generating an Encoded Password”](#) procedure.
2. Navigate to the %PUREWEB\_HOME%/webapp/WEB-INF/user-service.properties file.
3. Add entries in the file using the following format.  
<username>=<encodedpassword>,<roles>,enabled

**Example:**

```
janedoe=0DPiKuNIrrVmD8IUCuw1hQxNqZc=,ROLE_PUREWEB_USER,enabled
```

4. Save the changes before closing the file.
5. Restart the server for the changes to take effect.

End of procedure

**Changing Passwords**

**Purpose:** To change the password of a specific user by changing the base64 encoded SHA1 hash value.

Start of procedure

1. Generate a new password by following the steps in the [“Generating an Encoded Password”](#) procedure.
2. Navigate to the %PUREWEB\_HOME%/webapp/WEB-INF/user-service.properties file.
3. Replace the existing password found after the first equals symbol in the example below with the new encoded password.

**Example:**

```
janedoe=0DPiKuNIrrVmD8IUCuw1hQxNqZc=,ROLE_PUREWEB_USER,enabled
```

4. Save the changes before closing the file.
5. Restart the server for the changes to take effect.

---

Note: The password for the administrative user can also be changed by following the steps in the “[Changing the Administrator Password](#)” procedure.

---

## Generating an Encoded Password

Purpose: To generate an encode password for a PureWeb user.

### Start of procedure

1. Browse to a online encoding tool that uses a SHA1 algorithm to produce an encoded password in a base64 format. For example:  
[http://www.webutils.pl/SHA1\\_Calculator](http://www.webutils.pl/SHA1_Calculator) or <http://quickhash.com/>.
2. Use the tool to generate a new password.

---

Note: In an Linux environment, use the following command to generate passwords.

```
$ echo -n "new-password" | openssl dgst -sha1 -binary | openssl  
base64/v2IBCUBc9MKfPPQR9JBOIo8EbM=
```

---

### End of procedure

### Next Steps

- Use the encoded password when “[Adding PureWeb Users](#)” or “[Changing Passwords](#)” for existing users.

## Changing a Users Role

Purpose: To change a users role.

### Start of procedure

1. Navigate to the %PUREWEB\_HOME%/webapp/WEB-INF/user-service.properties file.
2. Edit the role(s) related to the specific user.

### Example:

```
janedoe=0DPiKuNIrrVmD8IUCuw1hQxNqZc=,ROLE_PUREWEB_USER,enabled
```

3. Save the changes before closing the file.
4. Restart the server for the changes to take effect.

End of procedure

## Disabling a User

**Purpose:** To disable access for a specific user.

**Start of procedure**

1. Navigate to the %PUREWEB\_HOME%/webapp/WEB-INF/user-service.properties file.
2. Edit the entry related to a specific user by changing enabled to disabled.

**Example:**

```
janedoe=ODPiKuNIrrVmD8IUCuw1hQxNqZc=,ROLE_PUREWEB_USER,disabled
```

3. Save the changes before closing the file.
4. Restart the server for the changes to take effect.

End of procedure

---

# LDAP Authentication Configuration

Sites that require LDAP authentication should add LDAP authentication provider definitions to the %PUREWEB\_HOME%/webapp/WEB-INF/pureweb-context.xml file and configure them appropriately.

## Configuring LDAP Authentication

**Purpose:** To configure the PureWeb® Server to authenticate users against one or more LDAP servers.

**Prerequisites**

- LDAP configuration requires site-specific information to identify the LDAP server URL, the manager DN and password, and the search base and filter values. This information should be available from the local LDAP administrator.

### Start of procedure

1. Navigate to the %PUREWEB\_HOME%/webapp/WEB-INF/pureweb-context.xml file.
2. Edit the example below to include your LDAP server information.
3. Edit the example below to map existing LDAP roles to their corresponding PureWeb® Server roles using the roleMap property of the ldapAuthenticationProvider to provide site-specific configuration without requiring changes to existing LDAP directories.

### Example:

```
<bean id="ldapAuthenticationProvider" class="pureweb.util.MappedLdapAuthenticationProvider">
  <property name="serverUrl" value="ldap://host:389"/>
  <property name="serverUsername" value="manager-user"/>
  <property name="serverPassword" value="manager-password"/>
  <property name="userSearchBase" value="OU=...,OU=...,CN=...,DC=...,DC=..."/>
  <property name="userSearchFilter" value="(name={0})"/>
  <property name="groupSearchBase" value="OU=...,OU=...,CN=...,DC=...,DC=..."/>
  <property name="groupSearchFilter" value="(member={0})"/>
  <property name="roleMap">
    <map>
      <entry key="ldapRole1" value="PUREWEB_SERVER_ADMIN"/>
      <entry key="ldapRole2" value="PUREWEB_SERVER_MONITOR"/>
      <entry key="ldapRole3" value="PUREWEB_USER"/>
    </map>
  </property>
</bean>
```

4. Save the file before closing.

### End of procedure

### Next Steps

- Follow the steps in the [“Registering the LDAP Authentication Provider”](#) procedure.

## Registering the LDAP Authentication Provider

**Purpose:** To explicitly register each LDAP authentication provider defined in the %PUREWEB\_HOME%/webapp/WEB-INF/pureweb-context.xml file with the authentication manager.

### Start of procedure

1. Navigate to the %PUREWEB\_HOME%/webapp/WEB-INF/pureweb-context.xml file.
2. Edit the example below to include line 79.

### Example:

```
74:<s:authentication-manager alias="authenticationManager" erase-credentials="false">
75:  <s:authentication-provider>
76:    <s:password-encoder ref="sha1Base64PasswordEncoder"/>
77:    <s:user-service properties="WEB-INF/user-service.properties"/>
78:  </s:authentication-provider>
79:  <s:authentication-provider ref="ldapAuthenticationProvider"/>
80:</s:authentication-manager>
```

3. Save the file before closing.

### End of procedure

---

**Note:** Custom plugins can also provide LDAP authentication by adding similar LDAP authentication provider definitions to the plugin xml files. Authentication providers defined in the plugin.xml configuration files will be automatically registered with the authentication manager via the authenticationManager property reference.

---

---

## Database Authentication Configuration

Sites that need to authenticate users against one or more database servers containing user information should add the appropriate data source and authentication provider definitions to the %PUREWEB\_HOME%/webapp/WEB-INF/pureweb-context.xml file for each database used as an authentication source.

### Configuring Database Authentication

**Purpose:** To configure the PureWeb® Server to authenticate users against one or more database servers.

### Prerequisites

- The specific database driver class, URL, username, password and queries for users (`usersByUsernameQuery`) and roles (`authoritiesByUsernameQuery`) is dependent on the details of the database used for authentication.

### Start of procedure

1. Navigate to the `%PUREWEB_HOME%/webapp/WEB-INF/pureweb-context.xml` file.
2. Edit the example below to include your database server information.
3. Edit the example below to map existing database roles to their corresponding PureWeb® Server roles using the `roleMap` property of the `JdbcAuthenticationProvider` to provide site-specific configuration without requiring changes to existing database roles.

### Example:

```
<bean id="jdbcAuthenticationProvider" class="pureweb.util.MappedJdbcAuthenticationProvider">
  <!-- optional passwordEncoder property if database password text is to be encrypted -->
  <property name="passwordEncoder" ref="sha1Base64PasswordEncoder"/>
  <property name="dataSource" ref="jdbcDataSource"/>
  <property name="usersByUsernameQuery"
    value="select username, password, enabled from users where username = ?"/>
  <property name="authoritiesByUsernameQuery"
    value="select username, role from userroles where username = ?"/>
  <property name="roleMap">
    <map>
      <entry key="jdbcRole1" value="PUREWEB_SERVER_ADMIN"/>
      <entry key="jdbcRole2" value="PUREWEB_SERVER_MONITOR"/>
      <entry key="jdbcRole3" value="PUREWEB_USER"/>
    </map>
  </property>
</bean>
```

4. Save the file before closing.

### End of procedure

### Next Steps

- Follow the steps in the [“Registering the Database Authentication Provider”](#) procedure.

## Registering the Database Authentication Provider

**Purpose:** To explicitly register each database authentication provider defined in the `%PUREWEB_HOME%/webapp/WEB-INF/pureweb-context.xml` file with the authentication manager.



### Start of procedure

1. Navigate to the %PUREWEB\_HOME%/webapp/WEB-INF/pureweb-context.xml file.
2. Edit the example below to include line 79.

### Example:

```
74: <s:authentication-manager alias="authenticationManager" erase-credentials="false">
75:   <s:authentication-provider>
76:     <s:password-encoder ref="sha1Base64PasswordEncoder"/>
77:     <s:user-service properties="WEB-INF/user-service.properties"/>
78:   </s:authentication-provider>
79:   <s:authentication-provider ref="jdbcAuthenticationProvider"/>
80: </s:authentication-manager>
```

3. Save the file before closing.

### End of procedure

---

Note: Custom plugins can also provide database authentication by adding similar JDBC authentication provider definitions to the plugin xml files. Authentication providers defined in plugin.xml configuration files are automatically registered with the authentication manager via the authenticationManager property reference.

---

---

## Reverse Proxies and Load Balancers

Below are the requirements for reverse proxies and load balancers.

### Requirements for Reverse Proxies

---

Notes: This does not cover the case of load balancing PureWeb® Server instances, only the case where a reverse proxy is pointing to a single instance of PureWeb®.

All server status pages depend on relative URLs that point to the base directory. If the reverse proxy modifies the path location of PureWeb®, you must translate all links in status pages.

Not all of the reverse proxies tested work in the standard connection mode provided by Microsoft Silverlight. In that case, you must use a connection mode instead.

---

In order for PureWeb® to work through reverse proxies, you must perform an analysis on the body of certain responses and translate the URLs to the proxied URL. You must translate the response to a POST request for the following URLs:

- /pureweb/app
- /pureweb/app/\*/\*/\*
- /pureweb/app/\*/\*/\*/\*
- /pureweb/share

You must parse the body of these responses to find links to the instance URLs. It is enough to look for `http://serverinstancehost:8080` for the host machine that is serving as a server instance.

All of the above URLs are of exact length. Do not translate any subdirectories below these URLs.

You only need to translate responses to POST requests, but you can translate others without problems.

## Recommendations for Load Balancers

PureWeb® has a very strong server affinity requirement. All requests need to go to a single server and you need to direct collaboration sessions to the same server. Do not have a single hostname point to a server farm. Instead, give each node of the server farm a unique hostname.

Your setup should have the following characteristics:

- Create a local hostname to each node of the farm.
- Create a corresponding hostname on the load balancer for each node. For simplicity, the load balancer's host name can be the same as the node's hostname.
- Configure the load balancer to direct all traffic from the same host to the same node in the farm. A secondary node may be setup as a backup. If the primary server goes down, bring the secondary node online.
- If the node in the farm responds to a different URL than the load balancer, configure the load balancer as a reverse proxy, see [“Requirements for Reverse Proxies”](#).

If load balancing is required, the load balancer may be set up with a virtual hostname used as a redirect server. Request flow will look like:

- The virtual host sends a request to the load balancer.
- The load balancer decides which farm node it wants to use to complete the session.
- The load balancer redirects the request to the selected node based on the node's hostname.
- The session continues as usual.

# 4

## Adding Applications to the PureWeb<sup>®</sup> Server

The PureWeb<sup>®</sup> Server is designed to support PureWeb<sup>®</sup>-enabled applications developed using the PureWeb<sup>®</sup> STK. Applications are installed into the server using the PureWeb<sup>®</sup> Server's Plugin Architecture.

The following sections will describe installation of an application including several conventions that all installed applications should follow.

This chapter contains the following sections:

- [Application Files, page 59](#)

---

### Application Files

Installation of a PureWeb<sup>®</sup>-enabled application may include files in several locations. An application may install the following:

- %PUREWEB\_HOME%/conf/<application name>-plugin.xml
- %PUREWEB\_HOME%/conf/<application name>-plugin.properties
- %PUREWEB\_HOME%/webapp/<application name>/
- %PUREWEB\_HOME%/webapp/themes/<application name>/
- %PUREWEB\_HOME%/webapp/WEB-INF/views/<application name>/

---

Note: You must reload plugins (or restart the PureWeb<sup>®</sup> Server) to make a newly installed application available.

---

Uninstallation of a PureWeb<sup>®</sup>-enabled application should remove all of the files that were added to the PureWeb<sup>®</sup> Server when the application was installed.

---

Note: You should stop the PureWeb® Server before removing an installed application.

---

## Registering your Application

**Purpose:** To register your application.

**Start of procedure**

1. Create a .xml file for your application using the “[Example of an <application name>-plugin.xml File](#)” as a reference.

The PluginPropertyLoader configured on line 15 loads application specific configuration properties. If no application specific configuration properties are required, the PluginPropertyLoader can be omitted.

Application startup information is specified in the ProcessFactory configuration in lines 19 to 44 and may make use of the configuration properties defined in the application's configuration properties file, see [Table 14](#).

Application client information is specified in the SupportedClients configuration in lines 46 to 58 and may make use of the configuration properties defined in the application's configuration properties file, see [Table 15](#).

### Example of an <application name>-plugin.xml File

```

01:<?xml version="1.0" encoding="UTF-8"?>
02:
03:<beans xmlns="http://www.springframework.org/schema/beans"
04:  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
05:  xsi:schemaLocation="http://www.springframework.org/schema/beans
06:    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
07:  import resource="security-config.xml"/>
08:
09:  <bean id="handlerMapping"
10:    class="org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping">
11:    <property name="useDefaultSuffixPattern" value="false"/>
12:    <property name="detectHandlersInAncestorContexts" value="false"/>
13:  </bean>
14:
15:  <bean class="pureweb.servlet.PluginPropertyLoader"/>
16:
17:  <bean id="licenseManager" class="pureweb.process.DefaultLicenseManager"/>
18:
19:  <bean id="<application name>ProcessFactory" class="pureweb.process.ProcessFactory">
20:    <property name="applicationRegistry" ref="applicationRegistry"/>
21:    <property name="licenseManager" ref="licenseManager"/>

```

```

22:     <property name="application" value="<application name>"/>
23:     <property name="description" value="The Application"/>
24:     <property name="directory" value="\${<application name>.home}"/>
25:     <property name="executable" value="\${<application name>.executable}"/>
26:     <property name="available" value="true"/>
27:     <property name="defaultProcess" value="\${<application name>.startup}"/>
28:     <property name="environment">
29:         <map>
30:             <entry key="HOME" value="\${<application name>.home}"/>
31:         </map>
32:     </property>
33:     <property name="options">
34:         <map>
35:             <entry key="--<application name>" value="true"/>
36:         </map>
37:     </property>
38:     <property name="arguments">
39:         <list>
40:             <value>/tmp</value>
41:             <value>/var</value>
42:         </list>
43:     </property>
44: </bean>
45:
46: <bean class="pureweb.cluster.SupportedClients" scope="prototype">
47:     <property name="supportedClients" ref="supportedClientsMap"/>
48:     <property name="application" value="example"/>
49:     <property name="clientMap">
50:         <map>
51:             <entry key="silverlight" value="example.xap"/>
52:             <entry key="ios"
53:                 value="http://itunes.apple.com/us/app/example/id398740007?mt=8&uo=4"/>
54:             <entry key="android" value="market://details?id=com.yourcompany.example"/>
55:         </map>
56:     </property>
57: </bean>
58:</beans>

```

2. Navigate to the %PUREWEB\_HOME%/conf/ directory.
3. Save the .xml file of your application.

Applications must follow the convention of naming their plugin .xml file with the application name as a prefix.

#### Example File Name:

%PUREWEB\_HOME%/conf/<application name>-plugin.xml

Use the properties in [Table 14](#) to create the process factory for the .xml file of your application.

**Table 14: Process Factory Configuration**

Name	Description	Optional / Mandatory	Valid Values
applicationRegistry	Defines the application registry that the process factory will register. If no application registry is defined, the process factory will not be available for use.	Mandatory	
licenseManager	Defines the license manager responsible for licensing of the specified application. The default license manager allows unrestricted access to the application. Third party application plugins can define their own license manager to handle custom licensing requirements.	Mandatory	
application	Defines the ID of the application that identifies the corresponding executable.	Mandatory	This should be unique across all applications defined on the server, or the PureWeb Server will error on startup and the plugin will not be loaded.
description	Defines the name of the application that will be displayed to users.	Mandatory	

**Table 14: Process Factory Configuration (Continued)**

Name	Description	Optional / Mandatory	Valid Values
directory	Defines the directory that the application will be started. Applications that need to be started in a particular directory must specify an appropriate value in this property.	Optional for applications that do not need to be started in a specific directory.	If not specified the working directory will be the PureWeb server's bin folder.
executable	Defines the full path to the application's main executable program. This program will be started with the specified environment, options and command line arguments when a request for the application is received.	Mandatory	
available	Controls the availability of the specified application. This can be used to disable the specified application so that it is not listed among available applications and cannot be requested by clients.	Optional	

**Table 14: Process Factory Configuration (Continued)**

Name	Description	Optional / Mandatory	Valid Values
defaultProcess	Controls startup of the specified application. One application may be configured as the default process and the PureWeb® Server will eagerly start instances of this process in all available user sessions to avoid delays associated with process startup. Only one default process may be defined, if more than one default process is defined only the first will be activated and subsequent process factory definitions will be ignored.	Optional	<ul style="list-style-type: none"> <li>• “true”—indicating that this is the default process.</li> <li>• “false”—indicating that this is <i>not</i> the default process.</li> </ul> <p>It is optional and does not have to be included in the configuration.</p>
environment	Defines additional environment variables required by the application executable.	Optional for applications that do not need additional environment variables.	<p>A map of key value pairs.</p> <p><b>Example:</b></p> <pre>&lt;property name="environment"&gt;   &lt;map&gt;     &lt;entry key="LD_LIBRARY_PATH" value="."/&gt;   &lt;/map&gt; &lt;/property&gt;</pre>



**Table 14: Process Factory Configuration (Continued)**

Name	Description	Optional / Mandatory	Valid Values
options	Defines the command line options that the application must be started with.	Optional for applications that do not need and specific command line options.	A map of key value pairs. <b>Example:</b> <pre>&lt;property name="options"&gt; &lt;map&gt; &lt;entry key="-c" value="server"/&gt; &lt;/map&gt; &lt;/property&gt;</pre>
arguments	Defines the command line arguments that the application must be started with, in the order that they must appear.	Optional for applications that do not require any specific command line arguments.	A list of values. <b>Example:</b> <pre>&lt;list&gt; &lt;value&gt;arg1&lt;/value&gt; &lt;value&gt;arg2&lt;/value&gt; &lt;/list&gt;</pre>

Use the properties in [Table 15](#) to configure supported clients for the .xml file of your application.

**Table 15: Supports Clients Configuration**

Name	Description	Optional/Mandatory
supportedClients	Defines the client map that this application will register with. If no client map is defined the application will not be available for use.	Mandatory
application	Defines the ID of the application that identifies the corresponding client.	Mandatory
clientMap	Defines the name of the client file (relative to the webapp folder) or a link to the client (for mobile clients) for each supported platform. At least one of “silverlight”, “flex”, “ios” and “android” must be specified. For types of “android” and “ios”, the value specified indicates a market URL to retrieve the client software when joining a collaborative session for the first time.	Mandatory

---

Note: A link to an iOS client on the App Store may be generated using Apple's Link Maker (<http://itunes.apple.com/linkmaker/>).

---

End of procedure

Next Steps

Follow the steps in the “[Configuring your Application](#)” procedure to configure your application.

## Configuring your Application

Purpose: To configure your application.

Start of procedure

1. Create a .properties file for your application using the “[Example <application name>-plugin.properties File](#)” as a reference.

Example <application name>-plugin.properties File

```
<application name>.home=/home/<application name>  
<application name>.executable=/home/<application name>/bin/<application name>
```

2. Navigate to the %PUREWEB\_HOME%/conf/ directory.
3. Save the .properties file for your application.

Applications must follow the convention of naming their plugin .properties file with the application name as a prefix.

Example File Name:

```
%PUREWEB_HOME%/conf/<application name>-plugin.properties
```

---

Note: Plugins are also configurable through the configuration management facilities available at <http://localhost:8080/pureweb/config/plugins> or through the Configuration link on the Status page. For more information see, “Configuration Management” on [page 29](#).

---

End of procedure

Next Steps

- If new resources are required, follow the steps in the “[Adding the Resources for your Application](#)” procedure.

## Adding the Resources for your Application

**Purpose:** To add new resources required by your application.

### Start of procedure

1. Navigate to the %PUREWEB\_HOME%/webapp directory.
2. Create a new directory (folder) to contain your resources.  
Applications should follow the convention of introducing any new resources in a directory named after the application to avoid collisions with other installed applications.

### Example:

%PUREWEB\_HOME%/webapp/<application name>

3. Add the resources for your application to the new directory.

### End of procedure

### Next Steps

- Protect the resources for your application from unauthorized access by following the steps in the [“Editing the security-config.xml File to Secure Resources”](#) procedure *or* by following the steps in the [“Creating Private Application Resources”](#) procedure.

## Editing the security-config.xml File to Secure Resources

**Purpose:** To limit access to the resources for your application by editing the security-config.xml file.

### Start of procedure

1. Navigate to the %PUREWEB\_HOME%/conf/security-config.xml file.
2. Edit the file by including a line of code that secures all of the resources for your application with a simple pattern that recursively matches your resource directory.

### Example:

```
<intercept-url pattern="/<application name>/**" access="ROLE_PUREWEB_USER"/>
```

3. Save the changes before closing the file.

### End of procedure

## Creating Private Application Resources

**Purpose:** To limit access to the resources for your application by making them private.

### Start of procedure

1. Navigate to %PUREWEB\_HOME%/webapp/WEB-INF/views/
2. Create a subdirectory.
3. Add the resources for your application.
4. Provide Controller implementations that make these resources available to users with the correct roles using the security annotations provided by the Spring Framework. This preferred approach allows applications to define their own security constraints without needing to make changes to low-level configuration files.

### End of procedure

### Next Steps

- If required, follow the steps in the [“Allocating the Graphics Processing Unit \(GPU\) for your Application”](#) procedure.

## Allocating the Graphics Processing Unit (GPU) for your Application

**Purpose:** To include an additional environment variable, DISPLAY, that is set when the application is started.

---

Note: This variable should *not* be listed in the environment property described in Table 14 on [page 62](#). The value contained in this variable will be the <display device> from the display.list configuration. For more information about display.list see, Table 9 on [page 43](#).

---

### Start of procedure

1. Follow the steps in the [“Configuring Display Devices”](#) procedure to specify which GPU has been allocated to the application instance.
2. Ensure that your application acquires resources on the specified GPU.

### End of procedure

## Displaying Links to Applications on the App Page

**Purpose:** To display links to applications on the App page by including the PluginLink bean to the plugin.xml file. For detailed information about the App page, see “Accessing Application Availability Information” on [page 23](#).

### Start of procedure

1. Navigate to the plugin.xml file.
2. Edit the file using the examples provided and [Table 16](#).

### Example PluginLink Bean

```
<!-- Adds the link under the Apps section of the PureWeb Web Application -->
<bean class="pureweb.process.PluginLink">
  <property name="registry" ref="pluginLinkRegistry"/>
  <property name="supportedClients" ref="supportedClients"/>
  <property name="name" value="Scribble App Advanced C#"/>
  <property name="description" value="A C# implementation of Scribble App Advanced"/>

  <!-- Optional. If specified the links generated will be as below. -->
  <!-- Comment out the next line to exclude this option. -->
  <property name="path" value="/pureweb/view?name=ScribbleAppAdvanced&client=silverlight"/>

  <property name="image" value="/themes/pureweb/default.png"/>
  <property name="role" value="ROLE_PUREWEB_USER"/>
</bean>
```

[Table 16](#) describes the PluginLink bean properties.

**Table 16: PluginLink Bean Properties**

Name	Description	Mandatory/Optional
registry	An implementation detail.	Mandatory
supportedClients	References the SupportedClients bean. Used to automatically generate links based on the available web clients.  If included, you must also add id="supportClients" to the SupportClients bean. See “Example SupportedClients Bean” on <a href="#">page 70</a> .  <b>NOTE:</b> To generate buttons for any specified web browser, do not specify a client (Flex or Silverlight).  <a href="#">Figure 16</a> shows how your application will appear on the App page if this optional property is included.	Optional
name	An application name displayed for the end-user.	Mandatory

**Table 16: PluginLink Bean Properties (Continued)**

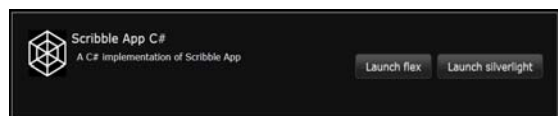
Name	Description	Mandatory/Optional
description	A detailed description of the application displayed for the end-user.	Mandatory
path	If provided this value will be used as the Launch link. Specifying this value overrides any automatic behavior. <a href="#">Figure 17</a> shows how your application will appear on the App page if this optional property is included.	Optional
image	The icon used to launch the application.	Mandatory
role	The role allowed to view the link when logged into the PureWeb® Server.	Mandatory

### Example SupportedClients Bean

```
<bean class="pureweb.cluster.SupportedClients" scope="prototype" id="supportedClients">
  <property name="supportedClients" ref="supportedClientsMap"/>
  <property name="application" value="ScribbleAppAdvanced"/>
  <property name="clientMap">
    <map>
      <entry key="silverlight" value="ScribbleAppAdvanced.xap"/>
      <entry key="flex" value="ScribbleClientAdvanced.swf"/>
      <entry key="ios" value=""/>
      <entry key="android" value=""/>
    </map>
  </property>
</bean>
```

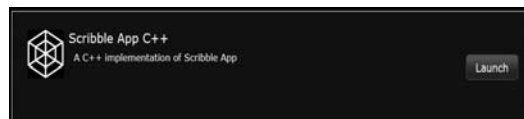
3. Save the file before closing.

[Figure 16](#) shows how the link to your application is displayed if you use the optional path property.



**Figure 16: Using the Optional supportedClients Property**

[Figure 17](#) shows how the link to your application is displayed if you use the optional path property.



**Figure 17: Using the Optional Path Property**

End of procedure





## Appendix

# A

## Secure Socket Layer (SSL) Certificate

This section is optional and used only for those sites that use SSL connections to access the PureWeb® Server.

This appendix contains the following sections:

- [About SSL Certificates, page 72](#)

---

## About SSL Certificates

SSL Certificates are required for the PureWeb® Server to support HTTPS connections from PureWeb® Clients. A certificate must be acquired from a recognized Certificate Authority or generated and configured in the Apache Tomcat container before HTTPS connections are enabled.

---

Note: Save the Key Store file in the %PUREWEB\_HOME%/tomcat/conf directory.

---



## Requesting and Installing an SSL Certificate

**Purpose:** SSL Certificates may be requested from many different internet providers all of which will follow a process similar to the following example.

### Start of procedure

---

Note: In this example, we use keytool, a key and certificate management utility. Keytool is included in the Java Development Kit (JDK).

---

1. Generate a Certificate. The Common Name must be the fully qualified domain name of the host to which the certificate is applied.

```
$ keytool -genkey -alias tomcat -keyalg RSA -keystore tomcat.keystore
```

2. Generate a Certificate Signing Request (CSR).

```
$ keytool -certreq -keyalg RSA -alias tomcat -file tomcat.csr -keystore tomcat.keystore
```

3. Submit the Certificate Signing Request to the Certificate Authority. Generally, this will involve copying and pasting the CSR generated above into an online enrollment form and possibly indicating that the server software is Apache Tomcat.
4. Receive the Signed Certificate from the Certificate Authority. This may be in the form of a zip file containing the Signed Certificate along with other certificates to form the Certificate Chain.
5. Import the Signed Certificate into a Key Store.

---

Note: Depending on which Certificate Provider that you used, the files that you receive may appear slightly different. Regardless, you *must* import *all* of the files that you receive.

---

```
$ keytool -import -alias cross -keystore tomcat.keystore -trustcacerts -file gd_cross_intermediate.crt
```

```
$ keytool -import -alias intermed -keystore tomcat.keystore -trustcacerts -file gd_intermediate.crt
```

```
$ keytool -import -alias tomcat -keystore tomcat.keystore -trustcacerts -file tomcat
```

6. Configure the Apache Tomcat container to use the Signed Certificate from the Key Store.

### End of procedure

### Next Steps

- See the “HTTP/HTTPS Connection Configuration” on [page 46](#) section for configuration of the Key Store and Certificate.

## Related Documentation Resources

The following resources provide additional information that is relevant to this software. Consult these additional resources as necessary.

### PureWeb® Server

- *PureWeb® Installation Guide for Microsoft Windows*, which provides detailed information on installing PureWeb® STK on a Microsoft Windows operating system.
- *PureWeb® iOS STK Installation Guide*, which provides detailed information on installing the PureWeb® iOS STK.
- *PureWeb® STK Quick Start Guide: Java*, which describes the sample applications and their code.
- *PureWeb® STK Quick Start Guide: C#*, which describes the sample applications and their code.
- *PureWeb® STK Quick Start Guide: C++*, which describes the sample applications and their code.
- *PureWeb® STK Quick Start Guide: Android*, which describes the sample applications and their code.
- *PureWeb® STK Quick Start Guide: Objective-C*, which describes the sample applications and their code.
- *PureWeb® Java Client STK API Reference*, which describes the Java Client STK.
- *PureWeb® Java Server STK API Reference*, which describes the Java Server STK.
- *PureWeb® Silverlight Client STK API Reference*, which describes the Silverlight Client STK.
- *PureWeb® C++ Server STK API Reference*, which describes the C++ Server STK.

Supplements :

- *PureWeb® DotNet Server STK API Reference*, which describes the C# Server STK.
- *PureWeb® STK Objective-C API Reference*, which describes the Objective-C STK.
- *PureWeb® Troubleshooting Guide*, which describes solutions to common issue.
- *PureWeb® STK Release Notes*.
- *PureWeb® STK Application Upgrading Notes*.

# Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

## Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

```
PW3.0_Java_QSG_03-2011_v3.0.001.00
```

You will need this number when you are talking with Calgary Scientific Support about this product.

## Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, may sometimes contain minor spelling, capitalization, or grammatical errors. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

## Type Styles

[Table 17](#) describes and illustrates the type conventions that are used in this document.

**Table 17: Type Styles**

Type Style	Used For	Examples
Italic	<ul style="list-style-type: none"> <li>Document titles</li> <li>Emphasis</li> <li>Definitions of (or first references to) unfamiliar terms</li> <li>Mathematical variables</li> </ul> <p>Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on <a href="#">page 77</a>).</p>	<p>Please consult the <i>Calgary Scientific Migration Guide</i> for more information.</p> <p>Do <i>not</i> use this value for this option.</p> <p>A <i>customary and usual</i> practice is one that is widely accepted and used within a particular industry or profession.</p> <p>The formula, <math>x + 1 = 7</math> where <math>x</math> stands for . . .</p>

**Table 17: Type Styles (Continued)**

Type Style	Used For	Examples
<p>Monospace font (Looks like teletype or typewriter text)</p>	<p>All programming identifiers and GUI elements. This convention includes:</p> <ul style="list-style-type: none"> <li>• The <i>names</i> of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages.</li> <li>• The values of options.</li> <li>• Logical arguments and command syntax.</li> <li>• Code samples.</li> </ul> <p>Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.</p>	<p>Select the Show variables on screen check box.</p> <p>In the Operand text box, enter your formula.</p> <p>Click OK to exit the Properties dialog box.</p> <p>Scribble service distributes the error messages in EventError events.</p> <p>If you select true for the inbound-bsns-calls option, all established inbound calls on a local agent are considered business calls.</p> <p>Enter exit on the command line.</p>
<p>Square brackets ([ ])</p>	<p>A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.</p>	<p>smcp_server -host [/flags]</p>
<p>Angle brackets (&lt;&gt;)</p>	<p>A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise.</p> <p><b>Note:</b> In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.</p>	<p>smcp_server -host &lt;confighost&gt;</p>

# Index

## Symbols

[] (square brackets)	77
< > (angle brackets)	77

## A

angle brackets	77
architecture	13
audience, for document	8

## B

brackets	
angle	77
square	77

## C

cluster	
accessing	21
configuration tasks	11
configuring	32
inactivating	27
managing	26
node properties	34
plugins	31
properties	32, 34
status	20, 28
testing	27
commenting on this document	9
conventions	
in document	76
type styles	76

## D

display device	
application	68

configuration	42
property	43
status	20
document	
audience	8
conventions	76
errors, commenting on	9
reading recommendations	8
version number	76

## F

font styles	
italic	76
monospace	77

## I

intended audience	8
italics	76

## L

license	2, 9, 15
logs	36
accessing	20, 25
configuration	36
descriptions	25
location	25
retention period	36

## M

monospace font	77
----------------	----

## P

password	17, 51
----------	--------

## Index

changing	21, 37
default	17
prerequisites	15
pureweb	
client	14
server	13
service	13
solution	13–14

## R

reading recommendations	8
roles	
collaborator	50
server administrator	50
server monitor	50
user	50
user admin	50

## S

securing	
application resources	67–68
applications	50
square brackets	77
support	
accessing information	24
contacting	9

## T

tasks	
application installation	12
cluster/node configuration	11
configuration	10
operational	9
ssl certificate tasks	12
type styles	
conventions	76
italic	76
monospace	77
typographical styles	76

## V

version numbering, document	76
-----------------------------	----