

OCTOBER 2019

Setup Guide

Web Services for Symphony 6.0.0



SirsiDynix®

All information contained herein is the proprietary property and trade secrets of SirsiDynix. This manual shall not be reproduced, transmitted, stored in a retrieval system, duplicated, used or disclosed in any form or by any means for any purpose or reason, in whole or in part, without the express written consent of SirsiDynix or as except provided by agreement with SirsiDynix. The information in this document is subject to change without notice and should not be construed as a commitment by SirsiDynix.

SirsiDynix grants the right of copying the enclosed material solely for the internal business use of the end user if (1) this document has been legitimately obtained by purchase or by license agreement in conjunction with SirsiDynix products, and (2) this copyright statement is included with each copy. All other copying or distribution is strictly prohibited. Complying with all applicable copyright laws is the responsibility of the user.

SirsiDynix trademarks include but are not limited to BLUEcloud™, BookMyne®, Directors Station®, EOS.web®, eResource Central®, MobileCirc®, SirsiDynix®, SirsiDynix Enterprise®, SirsiDynix Horizon®, SirsiDynix Portfolio™, SirsiDynix Symphony®, Unicorn®, Web Reporter™, and WorkFlows™. Unauthorized use of any SirsiDynix trademark is prohibited.

Other product and company names herein may be the trademarks of their respective owners and SirsiDynix claims no ownership therein. All titles, versions, trademarks, claims of compatibility, etc., of hardware and software products mentioned herein are the sole property and responsibility of the respective vendors. SirsiDynix makes no endorsement of any particular product for any purpose, nor claims responsibility for its operation and accuracy.

SirsiDynix products are developed exclusively at private expense. Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in DFARS 252.227-7013(b)(3) and in FAR 52.227-19(b)(1,2).

Contents

About this guide	vii
Summary of contents	vii
Possible differences between the software and this guide	viii
Documentation updates	viii
Comments and suggestions	ix
Getting Started	1
About Web Services for Symphony	1
System requirements	2
Hardware	3
Operating system	4
Java software	4
Apache Tomcat servlet container	4
Installing Web Services	6
Installation overview	6
Before you begin	7
Starting your installation	7
Running the installer	8
Changing the locale of the installer	9
Choosing the correct installation type	10
Installing Web Services with Tomcat	11
Installing Web Services only	13
Upgrading Web Services	15
Finding and setting Java environment variables	18
Installer configuration properties	19
Tomcat configuration properties	19

Symphony configuration properties	21
Web Services configuration properties	23
Troubleshooting installation	26
Advanced Installation	27
Creating the properties file	27
Essential properties	28
hat-session properties	32
admin-settings properties	36
Running the installer in Silent mode	37
Configuring Web Services	39
Web Services Admin Basics	40
Accessing the Admin console	40
Understanding the Admin console interface	41
Logging in to the Admin console	42
Logging out of the Admin console	43
Changing the admin username	43
Changing the admin password	44
Viewing the current status of Web Services	44
Viewing or updating a Web Services license key	45
Resetting Web Services caches	45
Fields: Status page	46
Updating ILS configuration options	48
Fields: ILS Configuration	48
Managing Client IDs	52
Fields: Client IDs	53
Configuring CAS single sign-on	57
Fields: Single Sign-on Setup	57
Fields: Create/Edit Single Sign-on URL	58
LDAP Setup	60
About LDAP Authentication	60
Managing LDAP Server Connections	61
Changing the LDAP Timeout Settings	62

Fields: LDAP Setup	63
Fields: Add/Edit LDAP Server	64
Patron Primary Address Configuration	67
Fields: Patron Primary Address Configuration	67
Resource Security Configuration	68
About the Resource Security Configuration	69
Setting resource security	69
Fields: /circulation/block	70
Fields: /circulation/block/detailLine	71
Fields: /user/patron	72
Fields: /user/patron/address1	76
Fields: /user/patron/address2	76
Fields: /user/patron/address3	76
Managing the log files	77
Viewing Web Services Logs	77
Changing the logging level	78
Fields: Logs	78
Customizing or localizing labels and messages	79
Uninstalling Web Services	81
Troubleshooting	82
General Troubleshooting	82
Restoring access if you cannot log in	83
Starting up Tomcat	83
Troubleshooting connection issues	84
Verifying that Web Services is running	85
Examining log files	86
catalina logs	86
requests log	86
symws log	87
BlackBox log	88
Troubleshooting common errors	89
BeanCreationException: Error creating bean	89

BindException: Address already in use: JVM_Bind	89
Connection refused - Cannot connect to host	90
Context initialization failed	90
HATIOException	90
HatIOException: Read of response timed out	91
HatSessionException: Unable to acquire session with server	91
Initializing session manager failed	92
Initial LDAP bind failed	92
listenerStart error	93
Appendix A: Key Concepts	94
Web Services base URL	94
Appendix B: Advanced Tomcat configuration	95
Appendix C: Managing password lockout	96
Configuring password lockout	96
Clearing the lockout cache	97
Fields: Lockout Settings	97
Appendix D: Configuring email Templates	100
Configuring Checkout Receipt Email Template	100
Configuring the Reset My PIN Template File	103
Index	105

About this guide

This *Setup Guide* explains how to install and set up Web Services for Symphony. It provides an overview of the software, including a brief summary of Web Services architecture, and describes system requirements and configuration properties.

This guide is intended for library system administrators who need to install and set up Web Services for Symphony for use with their SirsiDynix Symphony integrated library system. To use this guide, you should understand software administration and have a working knowledge of Java applications, Apache Tomcat in particular.

For more information about this guide, see the following topics:

Summary of contents	vii
Possible differences between the software and this guide	viii
Documentation updates	viii
Comments and suggestions	ix

Summary of contents

This guide contains these major sections:

- **Getting Started on page 1** provides overview information about Web Services.
- **Installing Web Services on page 6** gives instructions on how to install Web Services on your system using either a GUI or Console mode.
- **Advanced Installation on page 27** gives instructions on using the installer in Silent mode.
- **Configuring Web Services on page 39** describes the tasks associated with using the Web Services Admin console.
- **Uninstalling Web Services on page 81** gives instructions on how to uninstall Web Services from your system.
- **Troubleshooting on page 82** details different problem scenarios and how to fix them.
- **Appendix A: Key Concepts on page 94** details different Web Services concepts that are important when installing and configuring Web Services.
- **Appendix B: Advanced Tomcat configuration on page 95** details the recommended Tomcat settings and tells where to get information about how to configure advanced Tomcat settings.

Possible differences between the software and this guide

The names, labels, and sample windows in this guide reflect the default settings that are delivered with most new installations. The settings on your system may be different from these defaults, depending on your library's implementation choices and the way your system administrator sets up your system. For example, your system administrator can change labels and set up security to limit access to certain features.

Additionally, as you use the software, you can resize windows or customize your workspace. Consequently, your software environment may look and function differently than the environment described in the tasks in this guide.

This document is compatible with Web Services for Symphony 6.0.0. Information in this document may also be compatible with later versions.

Documentation updates

Updates to this guide are posted to the customer support website between releases, as necessary. These updates provide corrections to unclear, incorrect, or incomplete information. They also provide documentation for enhancements that were not complete at the time the guide was first published.

You may access the customer support website at <https://support.sirsiDynix.com>.



The customer support website requires a username and password. If you do not already have a username and password, contact your system administrator to receive one. If you are the system administrator for your library and need a username and password for the support website, please contact SirsiDynix Customer Support to receive one.

Documentation updates can include, but are not limited to, the following formats:

- PDF (Portable Document Format)
- HTML Webhelp
- EPUB
- Microsoft Word

To view a PDF file, you must install Adobe's Acrobat Reader on your workstation. You can download Acrobat Reader free of charge at Adobe's web site, <http://www.adobe.com>. Several open source eReaders are also available online which can open other formats such as EPUBs. Please contact SirsiDynix Customer Support if you are having trouble finding this guide in the format that you need.

Comments and suggestions

SirsiDynix welcomes and appreciates your comments on its documentation. We want to know what you think about our manuals and how we can make them better. If you have comments about this guide, please send them to docs@sirsidynix.com.

Be sure to include the title and version number of the guide and tell us how you used it. Then tell us your feelings about its strengths and weaknesses and any recommendations for improvements.

Getting Started

Welcome to Web Services for Symphony. You should read the entire guide before installing and setting up Web Services on your system. Also, make sure to read over the [Installation overview on page 6](#). The overview provides a Quick Steps Guide to the required steps for installing Web Services for Symphony on your system.

See the following topics for more information:

About Web Services for Symphony	1
System requirements	2
Hardware	3
Operating system	4
Java software	4
Apache Tomcat servlet container	4

About Web Services for Symphony

Web Services for Symphony is a Web application that provides simplified remote access to features of your SirsiDynix Symphony integrated library system (ILS). This information is used to power BLUEcloud Discovery, Staff, and Marketplace applications such as BLUEcloud Cataloging, eResource Central, BLUEcloud PAC, MobileCirc, and BookMyne. Web Services can also provide this information to clients across platforms or programming languages.

Web Services for Symphony 6.0.0 is compatible with the following SirsiDynix Symphony ILS versions:

- SirsiDynix Symphony 3.5.1, 3.5.1 SP1, 3.5.2, 3.5.2 SP1, or 3.5.3.

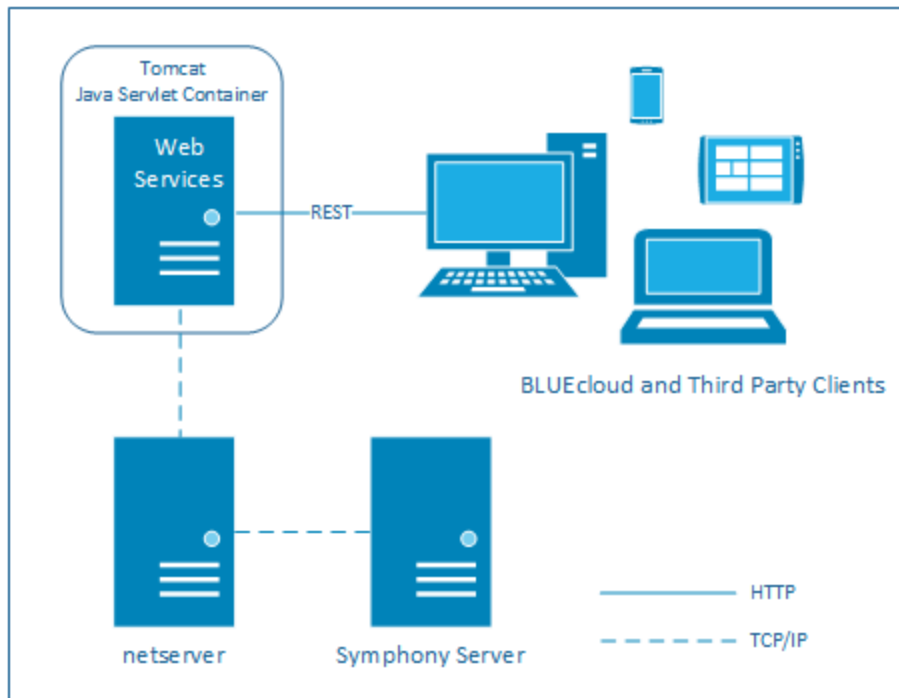
Note: SirsiDynix Symphony 3.5.1 is required for Web Services for Symphony to be fully functional. While it doesn't affect the current Symphony clients, such as MobileCirc, API developers will not be able to use these functions until they upgrade to 3.5.1 when it becomes available.

Web Services for Symphony 6.0.0 is compatible with SirsiDynix Enterprise or Portfolio 5.0 and greater. Web Services for Symphony 6.0.0 is not compatible with earlier versions of Enterprise or Portfolio.

For information about which version of Symphony, Enterprise, or Portfolio you are using, see the documentation for that software.

As a Java servlet, Web Services run in a Tomcat servlet container. Web Services can receive requests from both BLUEcloud and third-party clients, which in turn present that information to the library staff and patron users running the application. Web Services requires a Symphony netserver to communicate with the Symphony software. [Figure 1-1](#) illustrates the relationships of these components.

Figure 1-1: Web Services architecture



System requirements

Web Services for Symphony should be installed on a server-class system. For example, Web Services may be installed on the same server that runs your site's ILS server.

To use Web Services for Symphony, you must have the following software installed:

- Java SE Development Kit (JDK) or Java Runtime Environment (JRE) version 1.8u221 or greater (only updates within Java 8) or Amazon Corretto version 8.
- Apache Tomcat version 8.5 or higher (a distribution of Tomcat version 8.5.46 is included in the installer for your convenience). We recommend using the Tomcat version included with the installer for the best performance and compatibility with Web Services for Symphony.

Important: To ensure increased security and technological improvements, the next version of Web Services for Symphony will include Tomcat version 8.5.32. We strongly encourage you to prepare for this update. Detailed instructions for updating Tomcat to version 8.5.32 will be included with that version.

Beginning with version 8.5.32, Apache Tomcat has discontinued support of square brackets in URLs. If your system uses square brackets as it passes URLs, do not upgrade past 8.5.23 until you can make your code compliant with this and later versions of Tomcat.

Note: Remember to uninstall any previous versions of Tomcat before installing the latest version included with the installer.

See the following topics for more information:

Hardware	3
Operating system	4
Java software	4
Apache Tomcat servlet container	4

Hardware

Web Services for Symphony supports 64-bit architectures.

During typical operation, Web Services for Symphony requires 256 MB of free memory and about 250 MB of disk space. If you install multiple instances of Web Services, each instance will require an additional 100 MB of disk space for logging and about 100 MB of free memory.

By way of example, a site with 50 concurrent users will probably see memory use of 220–270 MB on the tomcat server and 200–300 MB memory usage on the ILS server. Free disk space on the tomcat server would need to be around 250 MB.

Although the default settings should be suitable for most installations, you may need to allow more memory or disk space, depending on your system's configuration. For example, if you are running Web Services with SirsiDynix Enterprise, you may need to increase the amount of memory available to Web Services to account for potentially heavy traffic.

In addition to memory and disk space, you can also configure heap space, stack space, perm space, and threading options in order to get the best performance with Web Services. For more information, see [Appendix B: Advanced Tomcat configuration on page 95](#).

Operating system

Web Services for Symphony is supported on the following operating systems:

- Microsoft Windows Server 2008 r2, 2012, 2012 r2, or 2016
- Linux (Red Hat Enterprise Linux 6 or 7; SUSE Linux Enterprise 10 or 11)
- Sun Solaris 10 or 11 (both Intel and Sparc)

Both 32- and 64-bit operating systems are supported.

Java software

Web Services for Symphony requires a Java Virtual Machine (JVM). The JVM can be either the Java SE Development Kit (JDK) or Java Runtime Environment (JRE).

Java 8

requires a Java Virtual Machine (JVM) version 1.8u221 or greater in 64-bit or Amazon Corretto version 8 depending on your hardware and operating system. The JVM can be either the Java SE Development Kit (JDK) or Java Runtime Environment (JRE).

Getting the latest version of Java

To download the required version of Java SE software, visit one of these sites:

Operating System	Site
Windows	http://www.oracle.com/technetwork/java/javase/downloads
Linux	http://www.oracle.com/technetwork/java/javase/downloads
Solaris	http://www.oracle.com/technetwork/java/javase/downloads
Amazon Web Services	https://docs.aws.amazon.com/corretto/latest/corretto-8-ug/downloads-list.html

Apache Tomcat servlet container

Web Services for Symphony requires Apache Tomcat 8.5 or higher in order to run.

For your convenience, a distribution of Tomcat 8.5.46 is included in the Web Services for Symphony installer (use the **Full Installation** option during installation).



We recommend using the Tomcat version included with the installer for the best performance and compatibility with Web Services for Symphony. Remember to uninstall any previous versions of Tomcat before installing the latest version that is included with the installer.

Beginning with version 8.5.32, Apache Tomcat has discontinued support of square brackets in URLs. If your system uses square brackets as it passes URLs, do not upgrade past 8.5.23 until you can make your code compliant with this and later versions of Tomcat.

If you do not use the Full Installation option, Tomcat must be installed prior to running the Web Services installer. You can download Tomcat at <http://tomcat.apache.org>. You may also need to make changes to your Tomcat installation for Web Services to run correctly (for more information, see [Appendix B: Advanced Tomcat configuration on page 95](#)).

Installing Web Services

This section explains how to install Web Services to run on your system. Please review the installation overview to become more acquainted with each of the required steps.

See the following topics for more information:

Installation overview	6
Before you begin	7
Starting your installation	7
Running the installer	8
Changing the locale of the installer	9
Choosing the correct installation type	10
Installing Web Services with Tomcat	11
Installing Web Services only	13
Upgrading Web Services	15
Finding and setting Java environment variables	18
Installer configuration properties	19
Tomcat configuration properties	19
Symphony configuration properties	21
Web Services configuration properties	23
Troubleshooting installation	26

Installation overview

Here are the essential steps that you need to take to install and run Web Services:

Task	Where to learn more
Review system requirements and gather required information.	Before you begin on page 7
Install Web Services.	Starting your installation on page 7
Configure Java variables (Linux or Solaris only).	Finding and setting Java environment variables on page 18
Start the Tomcat service (Windows) or Tomcat instance (Linux or Solaris).	Starting up Tomcat on page 83

Before you begin

Before you install Web Services for Symphony, review the hardware and software requirements listed in [System requirements on page 2](#). In particular, be sure you are using a valid version of the Java Virtual Machine (JVM).

It will also be helpful to gather the following information before running the installer (field names that you will encounter later in the installer are included in parentheses):

- The ILS Hostname and netserver Port for the netserver you will use to connect to your SirsiDynix Symphony ILS.
- A User ID for a Symphony user with administrator level privileges (Station User ID).
- A Policy name for a Station policy (Station Terminal).
- A Policy name for an Environment policy (Station Environment).
- A Policy name for a User Access policy with System Administrator level privileges (Station Access).
- The default language code that web services will use (Default language code).
- The time zone of your ILS server (Time Zone).
- The currency code for fines and payment transactions through web services (Currency).

For more information about these parameters, see [Symphony configuration properties on page 21](#) and [Web Services configuration properties on page 23](#).

Starting your installation

Web Services for Symphony may be installed on the following platforms:

- Windows
- Linux
- Solaris

During the installation process, you will specify configuration options. For more information about these options, see [Web Services configuration properties on page 23](#).

Before you can begin selecting options for your Web Services installation, you must first find and run the appropriate installer.

See the following topics for more information:

Running the installer

The file you use to launch the installer depends on the operating system that you are installing Web Services on. The following table lists the path to the installer file for each platform, where *<install source>* is the location of your installer files (for example, a CD-ROM drive or a directory where you unpacked the files, depending on how you received the software).

System	Installer to use
Windows	<code><install source>/Disk1/InstData/Windows/NoVM/symInst.exe</code>
Linux	<code><install source>/Disk1/InstData/Linux/NoVM/symInst.bin</code>
Solaris	<code><install source>/Disk1/InstData/Solaris/NoVM/symInst.bin</code>



The installer requires a valid version of the Java Virtual Machine (JVM) in order to run. For more information, see [Java software on page 4](#).



If you want to install a license at the same time as installing Web Services, please include the license file (.lic) in the same directory as the installer executable before running the installer.

The installer can be run in three different modes, depending on your system's configuration.

Graphical User Interface (GUI) Mode

If the installer detects that you are using a GUI-based operating system, it defaults to GUI mode. You can also force the installer to run in a GUI mode (for systems that have a GUI but default to console mode) using the `-i gui` command-line argument. For example:

```
symInst.exe -i gui (Windows) or symInst.bin -i gui (Linux or Solaris)
```

In GUI mode, you can click **Next** to move forward through the steps as you specify options. Click **Previous** to return to a previous step. You can also click **Cancel** to exit the installation process.

Console Mode

If the installer cannot display a GUI, it defaults to console mode. For command-line environments (Linux or Solaris), you can also force the installer to run in console mode using the `-i` console command-line argument. For example:

```
symInst.bin -i console
```

In console mode, you can press **Enter** to advance through the install process. Type **back** to return to a previous step. You can also press **Ctrl+C** to exit the installation process.

Silent Mode

For advanced users, the installer also supports running in silent mode. For more information about running the installer in silent mode, see [Running the installer in Silent mode on page 37](#).

Changing the locale of the installer

When running the installer in either GUI or console mode, the installer will launch with the default locale that is set for your system. The installer will also prompt you to select a locale immediately after it is launched. In GUI mode, this is represented by a dropdown with each locale that you can select. In console mode, you will be asked to specify the locale by choosing a number that corresponds to a list of locales in the console window. If you do not see the locale that you are looking for, review the list below to make sure that your locale is supported, then follow the directions for launching the installer with a specific locale.

To launch the installer in a different locale

1. Open a command prompt (Windows) or terminal window (Linux or Solaris).
2. Navigate to the directory on your system where the installer executable is located.
3. Enter one of the following commands, depending on your operating system and desired locale. See the list below for a list of supported locales along with their language code. The example below would run the installer using the Chinese-Traditional (`zh_TW`) locale.

```
symInst.exe -l zh_TW (Windows)
```

```
symInst.bin -l zh_TW (Linux or Solaris)
```

Supported Locales

The Web Services for Symphony installer currently supports the following locales:

- Chinese-Traditional (*zh_TW*)
- Chinese-Simplified (*zh_CN*)
- English (*en*)
- French-France (*fr*)
- French-Canadian (*fr_CA*)
- Spanish (*es*)

Choosing the correct installation type

If you are running the installer in GUI or console mode, the first steps of the installer will be the same until you are prompted to select an installation type. Please read about each of the installation types below and choose the type that matches your needs. You can find detailed instructions for each installation type within that type's section in this guide. If you can't decide which type of installation is appropriate for your system, contact SirsiDynix Customer Support.

- **Installing Web Services with Tomcat on page 11** — Use this installation type if you do not have Tomcat installed (Full Installation).
- **Installing Web Services only on page 13** — Use this installation type if you want to install additional instances of Web Services or if you want to install Web Services into your own installation of Tomcat (Web Services Only).

Important: If you choose this option, ensure that the existing Tomcat meets the requirements specified in **Apache Tomcat servlet container on page 4**.

- **Upgrading Web Services on page 15** — Use this installation type if you have an earlier version of Web Services and you want to replace it with the current version (Web Services Upgrade).

Important: If you choose this option, ensure that you have upgraded your Tomcat to meet the requirements specified in **Apache Tomcat servlet container on page 4**.

Installing Web Services with Tomcat

This section describes how to perform a full installation. Web Services for Symphony requires Apache Tomcat 8.5 or higher. For convenience, a distribution of Tomcat 8.5.46 is included with the installer. This installation type installs both Web Services and Tomcat.



(For Windows users) If you plan to install Tomcat to run as a service, you must run the installer as a system administrator.

To install Tomcat and Web Services

1. Review the information and requirements in [Before you begin on page 7](#).
2. Launch the installer. For more information, see [Running the installer on page 8](#).

Important: The installer requires a valid version of the Java Virtual Machine (JVM) in order to run. For more information, see [Java software on page 4](#).

3. Review and accept the terms of the SirsiDynix and third-party license agreements.

Note: Until you select **I accept the terms of the License Agreement**, the **Next** button will be inactive.

The software automatically scans for `java.exe` in expected locations.

4. Select the JVM instance to use for your Tomcat installation. If you do not select a JVM, the installer will use the first JVM from the list.

Note: For Linux and Solaris, the JVM that you select will be used for either the `JAVA_HOME` or `JRE_HOME` environment variable.

Note: In GUI mode, if you do not see the JVM you want, you can use one of these options to find `java.exe` on your system:

Option	Description
Search Another Location	Lets you specify another directory to scan. Any Java executables that are detected in this directory will be added to the list.

Option	Description
Choose Java Executable	Lets you select a specific Java executable. Note: If the <code>java.exe</code> you select does not meet the minimum version requirements, the installer will prompt you to choose again.

Note: In console mode, if you do not see the JVM you want, you can select the **Choose a Java VM already installed on this system** option to specify the path to another Java executable.

5. Choose a name for this instance of Web Services. This name is used in the Tomcat webapps directory where the application is installed. This name also appears in URLs used to access specific services, so only use alphanumeric characters that are valid in a URL (no spaces or other punctuation).
6. Specify the directory where you want to install Tomcat. Web Services is also installed into this directory.
7. Select the **Full Installation** option.
8. Specify the configuration details for Tomcat. For more information about specific properties, see [Tomcat configuration properties on page 19](#).
9. Specify the connection information for your Symphony ILS. For more information about specific properties, see [Symphony configuration properties on page 21](#).
10. Specify the configuration options for Web Services. For more information about specific properties, see [Web Services configuration properties on page 23](#).
11. Select the timezone for your Symphony ILS server. For more information about specific properties, see [Timezone on page 26](#).
12. Review the pre-installation summary. If you are satisfied, click **Install** (or press **Enter** in console mode) to proceed with the installation.
The installer completes the installation.
13. For Linux and Solaris, you need to set your `JAVA_HOME` or `JRE_HOME` environment variable to the JVM that you selected in step 4. For more information, see [Finding and setting Java environment variables on page 18](#).
14. Congratulations! You've finished installing Web Services for Symphony on your system. Before you can use Web Services, you need to start the Tomcat service (Windows) or Tomcat instance (Linux or Solaris). For more information, see [Verifying that Web Services is running on page 85](#).

Note: For most systems, the default configuration delivered in the Web Services installer will work without modification. However, depending on your system, there may be additional setup tasks you need to complete before running Web Services. For more information, see [Appendix B: Advanced Tomcat configuration on page 95](#).

Installing Web Services only

This section describes how to perform an installation that includes only Web Services. If you already have a version of Apache Tomcat installed that meets the minimum requirements (see [Apache Tomcat servlet container on page 4](#)), or if you need to install an additional instance of Web Services, then select this installation type.

If you choose to install multiple instances of Web Services, keep these guidelines in mind:

- Installing multiple instances of Web Services requires additional disk space for the service files, as well as space for logging. The additional space for these files totals approximately 150 MB for each instance of Web Services.
- Depending on the number of instances you install and the volume of requests Web Services receives, you may need to increase memory settings for Tomcat. For more information, see [Appendix B: Advanced Tomcat configuration on page 95](#).
- Web Services for Symphony 6.0.0 is backwards compatible with previous versions of Web Services. There is no need to install old versions of Web Services alongside version 6.0.0. To upgrade an existing Web Services instance, see [Upgrading Web Services on page 15](#).

To install Web Services only

1. Review the information and requirements in [Before you begin on page 7](#).
2. Launch the installer. For more information, see [Running the installer on page 8](#).

Important: The installer requires a valid version of the Java Virtual Machine (JVM) in order to run. For more information, see [Java software on page 4](#).

3. Review and accept the terms of the SirsiDynix and third-party license agreements.

Note: Until you select **I accept the terms of the License Agreement**, the **Next** button will be inactive.

The software automatically scans for `java.exe` in expected locations.

4. Select the JVM instance to use for your Tomcat installation. If you do not select a

JVM, the installer will use the first JVM from the list.

Note: For Linux and Solaris, the JVM that you select will be used for either the *JAVA_HOME* or *JRE_HOME* environment variable.

Note: In GUI mode, if you do not see the JVM you want, you can use one of these options to find `java.exe` on your system:

Option	Description
Search Another Location	Lets you specify another directory to scan. Any Java executables that are detected in this directory will be added to the list.
Choose Java Executable	Lets you select a specific Java executable. Note: If the <code>java.exe</code> you select does not meet the minimum version requirements, the installer will prompt you to choose again.

Note: In console mode, if you do not see the JVM you want, you can select the **Choose a Java VM already installed on this system** option to specify the path to another Java executable.

5. Choose a name for this instance of Web Services. This name is used in the Tomcat webapps directory where the application is installed. This name also appears in URLs used to access specific services, so only use alphanumeric characters that are valid in a URL (no spaces or other punctuation).

Important: If you install more than one instance of Web Services under the same Tomcat, then this name will need to be unique from any other instances of Web Services that are installed.

6. Specify the directory where Tomcat is installed. This is the directory where Web Services will be installed.
7. Select the **Web Services Only** option.
8. Specify the connection information for your Symphony ILS. For more information about specific properties, see [Symphony configuration properties on page 21](#).
9. Specify the configuration options for Web Services. For more information about specific properties, see [Web Services configuration properties on page 23](#).
10. Select the timezone for your Symphony ILS server. For more information about specific properties, see [Timezone on page 26](#).

11. Review the pre-installation summary. If you are satisfied, click **Install** (or press **Enter** in console mode) to proceed with the installation.

The installer completes the installation.

12. For Linux and Solaris, you need to set your `JAVA_HOME` or `JRE_HOME` environment variable to the JVM that you selected in step 4. For more information, see [Finding and setting Java environment variables on page 18](#).
13. Congratulations! You've finished installing Web Services for Symphony on your system. Before you can use Web Services, you need to start the Tomcat service (Windows) or Tomcat instance (Linux or Solaris). For more information, see [Verifying that Web Services is running on page 85](#).

Note: For most systems, the default configuration delivered in the Web Services installer will work without modification. However, depending on your system, there may be additional setup tasks you need to complete before running Web Services. For more information, see [Appendix B: Advanced Tomcat configuration on page 95](#).

Upgrading Web Services

This section describes how to perform an upgrade installation of Web Services. If you want to upgrade from an earlier version of Web Services for Symphony, you should select this installation type.



Before performing an upgrade, please make sure to back up the `admin-settings.properties` and `hat-session.properties` configuration files if you had made changes to them for an older version of Web Services. This will prevent you from losing any of your old settings when the installer overwrites these files during installation.



Depending on the version of Web Services that you are upgrading from, you may have to reconfigure your Client IDs following the upgrade. Please verify your Client IDs in the Web Services Admin console after upgrading. For more information, see [Managing Client IDs on page 52](#).

To upgrade Web Services

1. Shut down Tomcat.

Note: If you are running Tomcat as a service (Windows only), refer to the Windows documentation if you need information about stopping and starting services.

2. Review the information and requirements in [Before you begin](#) on page 7.
3. Launch the installer. For more information, see [Running the installer](#) on page 8.

Important: The installer requires a valid version of the Java Virtual Machine (JVM) in order to run. For more information, see [Java software](#) on page 4.

4. Review and accept the terms of the SirsiDynix and third-party license agreements.

Note: Until you select **I accept the terms of the License Agreement**, the **Next** button will be inactive.

The software automatically scans for `java.exe` in expected locations.

5. Select the JVM instance to use for your Tomcat installation. If you do not select a JVM, the installer will use the first JVM from the list.

Note: For Linux and Solaris, the JVM that you select will be used for either the `JAVA_HOME` or `JRE_HOME` environment variable.

Note: In GUI mode, if you do not see the JVM you want, you can use one of these options to find `java.exe` on your system:

Option	Description
Search Another Location	Lets you specify another directory to scan. Any Java executables that are detected in this directory will be added to the list.
Choose Java Executable	Lets you select a specific Java executable. Note: If the <code>java.exe</code> you select does not meet the minimum version requirements, the installer will prompt you to choose again.

Note: In console mode, if you do not see the JVM you want, you can select the **Choose a Java VM already installed on this system** option to specify the path to another Java executable.

6. Choose a name for this instance of Web Services. This name is used in the Tomcat

webapps directory where the application is installed. This name also appears in URLs used to access specific services, so only use alphanumeric characters that are valid in a URL (no spaces or other punctuation).

Important: You must specify the same instance name as when you previously installed Web Services, despite the default option of *symws*. For example, if you previously installed Web Services as *mylibraryws*, you need to change the name of the Web Services instance here to match that name. If you are unsure about your Web Services instance name, check your Tomcat webapps directory.

7. Specify the directory where Tomcat and Web Services are installed. This is the directory where the latest version of Web Services will be installed during the upgrade.
8. Select the **Web Services Upgrade** option.
9. Specify the connection information for your Symphony ILS. For more information about specific properties, see [Symphony configuration properties on page 21](#).
10. Specify the configuration options for Web Services. For more information about specific properties, see [Web Services configuration properties on page 23](#).
11. Select the timezone for your Symphony ILS server. For more information about specific properties, see [Timezone on page 26](#).
12. Review the pre-installation summary. If you are satisfied, click **Install** (or press **Enter** in console mode) to proceed with the installation.

The installer completes the installation.

13. For Linux and Solaris, you need to set your *JAVA_HOME* or *JRE_HOME* environment variable to the JVM that you selected in step 4. For more information, see [Finding and setting Java environment variables on page 18](#).
14. Congratulations! You've finished installing Web Services for Symphony on your system. Before you can use Web Services, you need to start the Tomcat service (Windows) or Tomcat instance (Linux or Solaris). For more information, see [Verifying that Web Services is running on page 85](#).

Note: For most systems, the default configuration delivered in the Web Services installer will work without modification. However, depending on your system, there may be additional setup tasks you need to complete before running Web Services. For more information, see [Appendix B: Advanced Tomcat configuration on page 95](#).

Finding and setting Java environment variables

To run Tomcat (Linux or Solaris), you must set either your *JAVA_HOME* or *JRE_HOME* variable to the correct path. Which variable you need to set depends on whether you selected a JDK or JRE Java executable for your JVM when you installed Web Services. Setting your Java environment variable correctly ensures that Tomcat uses the correct version of the JVM to run.



You do not need to set your Java environment variable if you are using Windows. The correct variable should have been selected automatically.

To find the current value for *JAVA_HOME* or *JRE_HOME*

1. Open a terminal window.
2. Enter one of the following commands, depending on whether you are using a Java JDK or JRE for your JVM:

```
env | grep JAVA
```

```
env | grep JRE
```

The system lists all environment variables that contain the string *JAVA* or *JRE*. This list will include values for *JAVA_HOME* or *JRE_HOME*, depending on which command you entered.

Once you have checked the value of *JAVA_HOME* or *JRE_HOME*, you may need to set one of them to the correct value. Depending on your system, you may want to create either a user- or system-level environment variable for *JAVA_HOME* or *JRE_HOME*. The instructions below show a common setup for a user-level environment variable. For additional details on setting user- and system-level environment variables, see the documentation for your operating system.

To set a user-level environment variable

1. Open a terminal window.
2. Enter one of the following commands, depending on whether you are using the Java JDK or JRE for your JVM:

```
export JAVA_HOME=/usr/java/jdk1.8u221/bin/java
```

```
export JRE_HOME=/usr/java/jre6/bin/java
```

The system sets the environment variable according to the path you specified.

Installer configuration properties

This section explains the configuration properties that you encounter while installing Web Services. Each property's name is provided along with the default value from the installer (where applicable) and a detailed description of how that property affects your Web Services installation.

See the following topics for more information:

Tomcat configuration properties

The following Tomcat properties should be specified when installing Web Services for Symphony. The default property values are valid for most installations. If the Tomcat HTTP, HTTPS, and shutdown ports you specify are in use by other services or applications, you will receive errors while running Web Services.

Property	Installer default	Description
Tomcat Service Name (Windows only)	tomcat8	<p>Specifies the name of the Tomcat service. The name should be short and include only letters and numbers (no spaces or other punctuation). The name that appears in the Windows Local Services Manager will be this name appended to "Apache Tomcat". For example, if my Web Services name was <i>mylibraryws</i>, I would find the service under <i>Apache Tomcat mylibraryws</i>.</p> <p>Important: This name must be unique. If the name you specify is already in use by another service, the installer will not be able to install Tomcat as a service.</p> <p>Note: If you leave Tomcat Service Name blank, Tomcat will not be installed as a service. You will need to start and stop it manually from the command line instead.</p>
Tomcat HTTP Port	8080	<p>Specifies the network port Tomcat will listen on for HTTP requests.</p> <p>Important: If the port you specify is already in use by another application, Tomcat will fail to start (see BindException: Address already in use: JVM_Bind on page 89). This port must be open to public access if using HTTP.</p>

Property	Installer default	Description
Tomcat HTTPS Port	8443	Specifies the network port Tomcat will listen on for HTTPS requests. Important: This port must be open to public access if using HTTPS.
Tomcat Shutdown Port	8009	Specifies the network port that Tomcat will listen on for shutdown requests.

Symphony configuration properties

The following Symphony connection properties should be specified when installing Web Services for Symphony. The default property values are valid for most installations. The various station values you specify allow Web Services to establish communication with the Symphony ILS.



All properties are required. If a property does not have a default value, you must supply a value appropriate to your ILS. If you do not specify valid values, you will receive errors while running Web Services.

Property	Installer default	Description
Host	no default	Specifies the IP address or host name of the Symphony ILS server.
Net Server Port	no default	Specifies the port number for connecting to the netserver.

Property	Installer default	Description
Station User ID	no default	<p>Specifies the SirsiDynix Symphony user to use to establish initial communication with the ILS. This needs to be a Symphony user ID that has administrator-level privileges. This can be an existing user ID or you can create a new user ID specifically for Web Services. For more information about user IDs and User Access in Symphony, refer to the WorkFlows online help.</p> <p>Note: This value is also used for unprivileged requests, such as searching or checking item availability.</p>
Station Terminal	PCGUI-DISP	<p>Specifies an ILS station policy. You can review valid values for this property in the Station wizard in the WorkFlows client.</p>
Station Clearance	NONE	<p>Specifies an Accountability policy. This policy is only used by systems where the Accountability module is enabled. You should leave this as <i>NONE</i> (or some other policy that inhibits access). If specific clearance is needed after Web Services starts, it will be determined from the clearance level of the user who is making requests.</p>

Property	Installer default	Description
Station Environment	SIRSI	Specifies an ILS environment policy. Sets the values such as preferred language and limit for sorting search results. You can review the default values for this property in the Environment wizard in the WorkFlows client.
Station Access	SIRSI	Sets the privileges for what commands can be executed by Web Services during initialization and for unprivileged requests. This needs to be a User Access policy that has administrator-level privileges. This can be an existing user access policy, or you can create a new user access policy specifically for Web Services. For more information about user IDs and User Access in Symphony, or about working with the User Access Wizard, refer to the WorkFlows online help.

Web Services configuration properties

The following Web Services properties should be specified when installing Web Services for Symphony. The default property values are recommended for most installations.

Connection Options

Property	Installer default	Description
Minimum Sessions	5	The minimum number of netserver sessions to keep open. You may want to increase this value if your Web Services instance handles a high volume of requests. This value must be a number.
Maximum Sessions	50	The maximum number of netserver sessions that Web Services will allow to be open. You may want to increase this value if your Web Services instance handles a high volume of requests. This value must be a number. Important: This value must be greater than or equal to the value of <i>Minimum Sessions</i> .
Max Session Idle Time	900	The time (in seconds) an open session to the netserver is allowed to remain idle before it is dropped and is no longer available for use. This value must be a number. Note: This only affects sessions that are above the number of minimum sessions specified.

Property	Installer default	Description
Session Time Out	30000	The maximum amount of time (in milliseconds) that Web Services processes will wait for responses to return from the netserver before timing out. This value must be a number.
Keep Alive Time	0	How often (in minutes) that Web Services will send keep alive messages to maintain connections between clients. The default value is 0, which turns off keep alive messages. This value must be a number.

General Options

Property	Installer default	Description
Default language code	en	Specifies the default language code for this instance of Web Services. Messages that are sent from Web Services will be localized according to the code specified here. This value can be any valid ISO-639-1 two-character language code.
Currency	no default	Specifies the default currency code that will be used by this instance of Web Services for fines and payment requests. This value can be any valid ISO 4217 currency code.

Property	Installer default	Description
Logging Prefix	<web application name>	Specifies the prefix to use on log files for this instance of Web Services. The default prefix is the name you specified for this instance of Web Services.
Logging Directory	../logs	Specifies the directory where Web Services will store any log files that it generates. By default, this directory is placed in the Tomcat directory.

Timezone

Specifies the time zone of the Symphony instance, which the ILS includes whenever it returns datetime data. The datetime for the client is controlled by Web Services. The time stamp is converted on incoming requests to the Web Services time zone, and the system assumes that outgoing responses are in the same time zone.

Troubleshooting installation

In Linux and Solaris, installer messages go to stdout, so they appear in the terminal window you use to launch the installer if you start from a command line.

In Windows, installer messages are displayed in the installer window or in dialog boxes.


On completion, the installer also creates an installation log on all platforms in the target Tomcat directory. The name of this log file includes the name you specified for your instance of Web Services.

If you encounter errors during installation, refer to these resources.

Advanced Installation

This section explains how to install Web Services using silent mode. This installation mode allows you to configure an `installer.properties` file in advance that the installer uses to perform the installation for your system. When the installer is run in silent mode, there is no console or GUI display, and you do not have to step through the installer.

This functionality can be very helpful for libraries that need to routinely perform a high number of installations. After configuring the `installer.properties` file for each instance of Web Services, the installations can be scripted to run together as a batch.

 Because the silent mode installation requires a more advanced knowledge of each of the settings used to configure Web Services, this installation mode is not recommended for most installations. If you still have additional questions about silent mode installation after reading this section, contact SirsiDynix Customer Support.

See the following topics for more information:

Creating the properties file	27
Essential properties	28
hat-session properties	32
admin-settings properties	36
Running the installer in Silent mode	37

Creating the properties file

Before you can run the installer in silent mode, you must first create an `installer.properties` file that contains the settings and options the installer will use. This section of the guide explains the different sections that can be included in the `installer.properties` file and the details which are required, depending on the type of installation you are trying to perform (Full Installation, Web Services Only, or Web Services Upgrade).



Templates of the `installer.properties` file are available on the SirsiDynix Support Center website for download. These templates are pre-populated with useful examples. We recommend downloading these templates and reviewing them along with this guide. For additional support in using these templates, please contact SirsiDynix Customer Support.

See the following topics for more information:

Essential properties

The following properties must be included in the `installer.properties` file for all installation types. A table is provided which describes each of the properties shown in the example. When multiple lines of the same property are included, some will be commented out (shown with a `#` at the beginning of the line). This indicates that a choice must be made between a set of possible options for that property (all possible options are included).

Required properties for all installation types

```
#Choose Installer Mode (Optional)
#-----
INSTALLER_UI=SILENT
#INSTALLER_UI=CONSOLE
#INSTALLER_UI=GUI

#Choose Properties Load Configuration Settings
#-----
LOAD_DEFAULT=FALSE
LOAD_FROMFILE=TRUE
LOAD_EXISTING=TRUE

#Select Java Virtual Machine
#-----
JDK_HOME=
JAVA_DOT_HOME=
JAVA_EXECUTABLE=

#Web Services Instance Name
#-----
WAR_NAME=

#Tomcat Installation Directory
#-----
USER_INSTALL_DIR=

#Choose Installation Type
#-----
#For full installation of both Web Services and the included Tomcat
```

```

CHOSEN_FEATURE_LIST=Tomcat,WS Install
CHOSEN_INSTALL_FEATURE_LIST=Tomcat,WS Install
#For installation of a new Web Services instance in an existing Tomcat
#CHOSEN_FEATURE_LIST=WS Install
#CHOSEN_INSTALL_FEATURE_LIST=WS Install
#For upgrading an existing Web Services instance
#CHOSEN_FEATURE_LIST=WS Upgrade
#CHOSEN_INSTALL_FEATURE_LIST=WS Upgrade

#Configure Tomcat
#-----
tomcat.service.name=tomcat8
tomcat.http.port=8080
tomcat.redirect.port=8443
tomcat.shutdown.port=8009
    
```

Property Descriptions

Property	Description
INSTALLER_UI	Specifies the mode that the installer will run in. This includes the following options: <ul style="list-style-type: none"> • SILENT • CONSOLE • GUI <p>Note: This option should be left as <i>SILENT</i> in order to use the silent install mode.</p>
LOAD_DEFAULT	Indicates whether the installer will load the default properties values that are included with the installer. For more information about the default properties values included with the installer, see Installer configuration properties on page 19 . This value should be left as <i>FALSE</i> while using the silent install mode. <p>Note: This value will be considered <i>TRUE</i> if it is not set to <i>FALSE</i>.</p>
LOAD_FROMFILE	Indicates whether the installer will load properties values from configuration files that are included in the installer executable file's directory. This value should be set to <i>FALSE</i> if no such configuration files will be used or <i>TRUE</i> if they will be used. <p>Note: This value will be considered <i>TRUE</i> if it is not set to <i>FALSE</i>.</p>

Property	Description
LOAD_EXISTING	<p>Indicates whether the installer will load properties values from existing configuration files in the specified Tomcat directory (represented in the properties file by <i>USER_INSTALL_DIR</i> and <i>WAR_NAME</i>). This value should be set to <i>FALSE</i> if existing configuration files either do not exist or you do not want to use them.</p> <p>Note: This value will be considered <i>TRUE</i> if it is not set to <i>FALSE</i>.</p>
JDK_HOME	<p>Specifies the path to the Java root directory if you are using the Java JDK for your JVM.</p> <p>For example:</p> <ul style="list-style-type: none"> • <i>JDK_HOME=C:\Program Files\Java\jdk1.8u221</i> (Windows) <p><i>JDK_HOME=/usr/java/jdk1.8u221/bin/java</i> (Linux or Solaris)</p> <p>This property corresponds to the <i>JAVA_HOME</i> environment variable.</p> <p>Note: You only need to define <i>JDK_HOME</i> if you are using the Java JDK for your JVM.</p>
JAVA_DOT_HOME	<p>Specifies the path to the Java root directory if you are using the Java JRE for your JVM.</p> <p>For example:</p> <ul style="list-style-type: none"> • <i>JDK_HOME=C:\Program Files\Java\jdk1.8u221\jre</i> (Windows) <p><i>JDK_HOME=/usr/java/jre6/bin/java</i> (Linux or Solaris)</p> <p>This property corresponds to the <i>JRE_HOME</i> environment variable.</p> <p>Note: You only need to define <i>JAVA_DOT_HOME</i> if you are using the Java JRE for your JVM.</p>
JAVA_EXECUTABLE	<p>Specifies the path to the Java executable. This will commonly be <i>JRE_HOME\bin\java</i> or <i>JAVA_HOME\bin\java</i>, depending on whether you are using the Java JDK or JRE as your JVM.</p>

Property	Description
WAR_NAME	<p>Specifies a name for this instance of Web Services. This name is used in the Tomcat webapps directory where the application is installed and is also the name that will appear in URLs used to access specific services, so only use alphanumeric characters that are valid in a URL (no spaces or other punctuation).</p> <p>For example:</p> <pre>#WAR_NAME=symws</pre>
USER_INSTALL_DIR	<p>Specifies the directory where Tomcat will be installed (for a full installation) or where Tomcat already exists (for installing another instance of Web Services or upgrading).</p> <p>For example:</p> <ul style="list-style-type: none"> • <i>USER_INSTALL_DIR=C:\Program Files\SirsiDynix\WebServices</i> (Windows) • <i>USER_INSTALL_DIR=~/.SirsiDynix/WebServices</i> (Linux or Solaris)
CHOSEN_FEATURE_LIST	<p>In combination with <i>CHOSEN_INSTALL_FEATURE_LIST</i>, specifies the installation type. Both properties must be included and are already grouped according to the different types of installation you can perform. Be sure to remove the comment markers (#) from both lines for your desired installation type, and then comment the lines of the installation types that you will not be using.</p>
CHOSEN_INSTALL_FEATURE_LIST	<p>In combination with <i>CHOSEN_FEATURE_LIST</i>, specifies the installation type. Both properties must be included and are already grouped according to the different types of installation you can perform. Be sure to remove the comment markers (#) from both lines for your desired installation type, and then comment the lines of the installation types that you will not be using.</p>

Property	Description
<code>tomcat.service.name</code> (Windows only)	<p>Specifies the name of the Tomcat service. The name should be short and include only letters and numbers (no spaces or other punctuation). The name that appears in the Windows Local Services Manager will be this name appended to "Apache Tomact". For example, if my Web Services name was <i>mylibraryws</i>, I would find the service under <i>Apache Tomcat mylibraryws</i>.</p> <p>Important: This name must be unique. If the name you specify is already in use by another service, the installer will not be able to install Tomcat as a service.</p> <p>Note: If you leave Tomcat Service Name blank, Tomcat will not be installed as a service. You will need to start and stop it manually from the command line instead.</p>
<code>tomcat.http.port</code>	<p>Specifies the network port Tomcat will listen on for HTTP requests.</p> <p>Important: If the port you specify is already in use by another application, Tomcat will fail to start (see BindException: Address already in use: JVM_Bind on page 89). This port must be open to public access if using HTTP.</p>
<code>tomcat.redirect.port</code>	<p>Specifies the network port Tomcat will listen on for HTTPS requests.</p> <p>Important: This port must be open to public access if using HTTPS.</p>
<code>tomcat.shutdown.port</code>	<p>Specifies the network port that Tomcat will listen on for shutdown requests.</p>

hat-session properties

The following properties should be included in the `installer.properties` file only if both `LOAD_FROMFILE` and `LOAD_EXISTING` are set to `FALSE` (for more information, see **Essential properties** on page 28). Otherwise, they are optional, as they will be overwritten by the settings found in these files.

hat-session properties

```
#hat-session.properties values
#-----
hat_host=
hat_port=
hat_stationUserID=
hat_stationTerminalName=PCGUI-DISP
hat_stationLoginClearance=NONE
hat_stationLoginEnvironment=SIRSI
hat_stationLoginUserAccessCode=SIRSI
hat_minSessions=5
hat_maxSessions=50
hat_maxIdleTime=900
hat_socketTimeOut=30000
hat_keepAliveTime=0
hat_defaultLanguage=en
hat_timezoneID=
hat_currency=
```

Property Descriptions

Property	Description
hat_host	Specifies the IP address or host name of the Symphony ILS server.
hat_port	Specifies the port number for connecting to the netserver.
hat_stationUserID	<p>Specifies the SirsiDynix Symphony user to use to establish initial communication with the ILS. This needs to be a Symphony user ID that has administrator-level privileges. This can be an existing user ID or you can create a new user ID specifically for Web Services. For more information about user IDs and User Access in Symphony, refer to the WorkFlows online help.</p> <p>Note: This value is also used for unprivileged requests, such as searching or checking item availability.</p>

Property	Description
hat_stationTerminalName	Specifies an ILS station policy. You can review valid values for this property in the Station wizard in the WorkFlows client.
hat_stationLoginClearance	Specifies an Accountability policy. This policy is only used by systems where the Accountability module is enabled. You should leave this as <i>NONE</i> (or some other policy that inhibits access). If specific clearance is needed after Web Services starts, it will be determined from the clearance level of the user who is making requests.
hat_stationLoginEnvironment	Specifies an ILS environment policy. Sets the values such as preferred language and limit for sorting search results. You can review the default values for this property in the Environment wizard in the WorkFlows client.
hat_stationLoginUserAccessCode	Sets the privileges for what commands can be executed by Web Services during initialization and for unprivileged requests. This needs to be a User Access policy that has administrator-level privileges. This can be an existing user access policy, or you can create a new user access policy specifically for Web Services. For more information about user IDs and User Access in Symphony, or about working with the User Access Wizard, refer to the WorkFlows online help.

Property	Description
hat_minSessions	<p>The minimum number of netserver sessions to keep open. You may want to increase this value if your Web Services instance handles a high volume of requests. This value must be a number.</p>
hat_maxSessions	<p>The maximum number of netserver sessions that Web Services will allow to be open. You may want to increase this value if your Web Services instance handles a high volume of requests. This value must be a number.</p> <p>Important: This value must be greater than or equal to the value of <i>Minimum Sessions</i>.</p>
hat_maxIdleTime	<p>The time (in seconds) an open session to the netserver is allowed to remain idle before it is dropped and is no longer available for use. This value must be a number.</p> <p>Note: This only affects sessions that are above the number of minimum sessions specified.</p>
hat_socketTimeout	<p>The maximum amount of time (in milliseconds) that Web Services processes will wait for responses to return from the netserver before timing out. This value must be a number.</p>
hat_keepAliveTime	<p>How often (in minutes) that Web Services will send keep alive messages to maintain connections between clients. The default value is 0, which turns off keep alive messages. This value must be a number.</p>

Property	Description
hat_defaultLanguage	Specifies the default language code for this instance of Web Services. Messages that are sent from Web Services will be localized according to the code specified here. This value can be any valid ISO-639-1 two-character language code.
hat_timezoneID	Specifies the time zone of the Symphony instance, which the ILS includes whenever it returns datetime data. The datetime for the client is controlled by Web Services. The time stamp is converted on incoming requests to the Web Services time zone, and the system assumes that outgoing responses are in the same time zone. Note: The value for your time zone can be any valid Java time zone ID. Commonly, these IDs are a continent followed by a city name, for example, <i>America/New_York</i> or <i>Australia/Sydney</i> .
hat_currency	Specifies the default currency code that will be used by this instance of Web Services for fines and payment requests. This value can be any valid ISO 4217 currency code.

admin-settings properties

The following properties should be included in the `installer.properties` file only if both `LOAD_FROMFILE` and `LOAD_EXISTING` are set to `FALSE` (for more information, see [Essential properties on page 28](#)). Otherwise, they are optional, as they will be overwritten by the settings found in these files.

admin-settings properties

```
#admin-settings.properties values
```

```
#-----
#It is recommended that the admin_logging.prefix match the WAR_NAME
admin_logging.prefix=
admin_logging.directory=../logs
```

Property Descriptions

Property	Description
admin_logging.prefix	Specifies the prefix to use on log files for this instance of Web Services. The default prefix is the name you specified for this instance of Web Services.
admin_logging.directory	Specifies the directory where Web Services will store any log files that it generates. By default, this directory is placed in the Tomcat directory.

Running the installer in Silent mode

The file you use to launch the installer depends on the operating system that you are installing Web Services on. The following table lists the paths to the installer files, where *<install source>* is the location of your installer files (for example, a CD-ROM drive or a directory where you unpacked the files, depending on how you received the software).

System	Installer to use
Windows	<install source>/Disk1/InstData/Windows/NoVM/symInst.exe
Linux	<install source>/Disk1/InstData/Linux/NoVM/symInst.bin
Solaris	<install source>/Disk1/InstData/Solaris/NoVM/symInst.bin



The installer requires a valid version of the Java Virtual Machine (JVM) in order to run. For more information, see [Java software on page 4](#).



If you want to install a license at the same time as installing Web Services, please include the license file (.lic) in the same directory as the installer executable before running the installer.

To run the installer in Silent mode

1. Prepare your `installer.properties` file according to the directions in [Creating the properties file on page 27](#), and put the `installer.properties` file in the same directory as the installer executable.
2. Locate the correct installer executable using the information listed above.
3. Open a command prompt or terminal window and navigate to the directory where the installer executable is located.
4. If you set `LOAD_FROMFILE` to `TRUE` in your `installer.properties` file (for more information, see [Essential properties on page 28](#)), ensure that you have put the necessary configuration files in the same directory as the `installer.properties` file and the installer executable.
5. Run the installer by entering one of the commands below, depending on your operating system.

`symInst.exe` (Windows)

`symInst.bin` (Linux or Solaris)

Configuring Web Services

This section will introduce you to the basics of the Web Services Admin console and describe the configuration and management tasks that you might need to perform.

See the following topics for more information:

Web Services Admin Basics	40
Accessing the Admin console	40
Understanding the Admin console interface	41
Logging in to the Admin console	42
Logging out of the Admin console	43
Changing the admin username	43
Changing the admin password	44
Viewing the current status of Web Services	44
Viewing or updating a Web Services license key	45
Resetting Web Services caches	45
Fields: Status page	46
Updating ILS configuration options	48
Fields: ILS Configuration	48
Managing Client IDs	52
Fields: Client IDs	53
Configuring CAS single sign-on	57
Fields: Single Sign-on Setup	57
Fields: Create/Edit Single Sign-on URL	58
LDAP Setup	60
About LDAP Authentication	60
Managing LDAP Server Connections	61
Changing the LDAP Timeout Settings	62
Fields: LDAP Setup	63
Fields: Add/Edit LDAP Server	64
Patron Primary Address Configuration	67
Fields: Patron Primary Address Configuration	67
Resource Security Configuration	68
About the Resource Security Configuration	69
Setting resource security	69

Fields: /circulation/block	70
Fields: /circulation/block/detailLine	71
Fields: /user/patron	72
Fields: /user/patron/address1	76
Fields: /user/patron/address2	76
Fields: /user/patron/address3	76
Managing the log files	77
Viewing Web Services Logs	77
Changing the logging level	78
Fields: Logs	78
Customizing or localizing labels and messages	79

Web Services Admin Basics

The Web Services for Symphony Admin console allows you to configure and manage an instance of Web Services. This section describes the basic functions and concepts for the Web Services Admin console.

See the following topics for more information:

Accessing the Admin console

The Web Services for Symphony Admin console is a Web application that you can use from anywhere that you have access to the internet.

The URL for accessing the Admin console is

```
<service-instance>/admin
```


where *<service-instance>* is the base URL for an instance of Web Services. For more information, see [Web Services base URL on page 94](#).

For example, if your Tomcat host name is *libraryapps.example.org*, and you used the default port (8080), and the default Web Services name (*symws*), the base URL for your Web Services instance would be:

```
http://libraryapps.example.org:8080/symws/admin
```

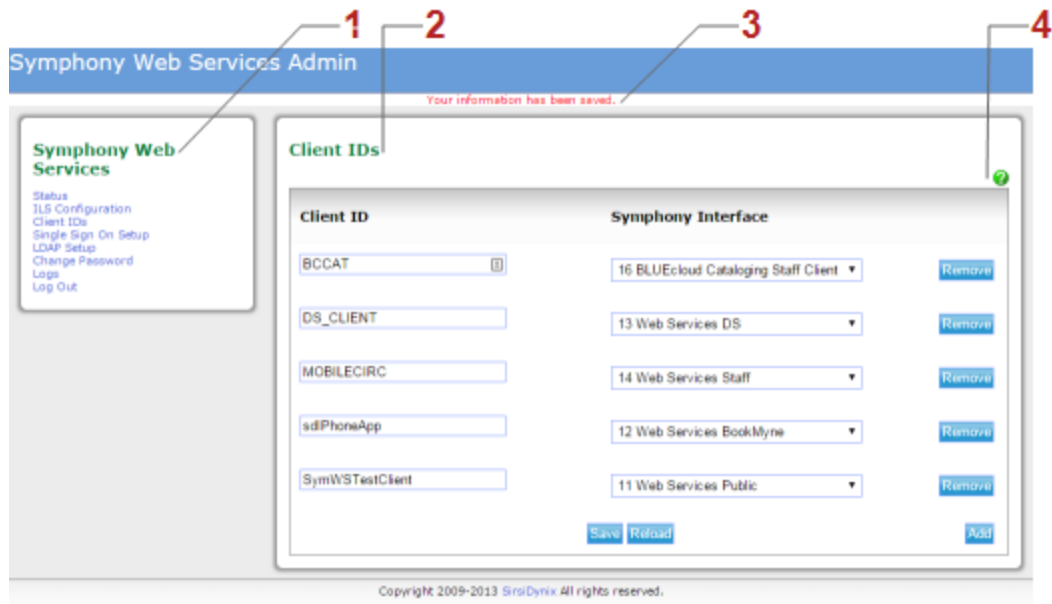
Please be aware of these browser requirements for using the Admin console:

Browser	Supported Versions	Platforms
Chrome	68, 69	Windows/Macintosh
Edge	41.16299.15, 42.17134	Windows 10
Firefox	61, 62	Windows/Macintosh
Internet Explorer	10, 11	Windows
Safari	10.1.2, 11.1.2	Macintosh

 As a Web application, the Admin console uses JavaScript and cookies. If either JavaScript or cookies is disabled, some features will not work.

Understanding the Admin console interface

Here is a sample of the Web Services Admin console interface with a description of some of the key features referred to in this guide.



Navigation pane

When you log in to the Admin console, a navigation pane (1) shows the different areas of the Admin console. Click on an option to view or configure settings.


Workspace

When you click an option in the Navigation pane, the software displays additional settings or options in the Workspace area (2). Use the buttons, controls, and form fields to view or modify settings.

Message bar

When you perform an action in the Admin console (for example, saving configuration changes), the software displays a message to indicate success or failure (3). If the operation failed, the message may include information about the cause of the failure.

Help icon

A question mark icon  indicates that there is help available (4). Click on the icon to open the online help and view information about the options you see.

Logging in to the Admin console

You must log in to use the Web Services Admin console.

When you first install Web Services, the default username and password for the admin console are both *admin*. Be sure to change the password to prevent unauthorized access (for more information, see [Changing the admin password on page 44](#)). You can also change the Admin console username if you want to (for more information, see [Changing the admin username on page 43](#)).

The username and password fields are case sensitive. If you forget your username or password, see [Restoring access if you cannot log in on page 83](#).

To log in to the Admin console

1. Open the Log In page. For more information, see [Accessing the Admin console on page 40](#).
2. Type the Admin console username and password.
3. Click **Log In**.

If the username and password are correct, the software starts a new session for you and displays the Status page.

Important: Be sure to log out when you are finished using the Admin console to prevent unauthorized access. For more information, see [Logging out of the Admin console on page 43](#).

Logging out of the Admin console

For your security, if you do not perform any actions in Admin console for some time, the server will log you out. However, you should log out of the Admin console yourself when you finish configuring Web Services. This precaution will help to prevent unauthorized access.



Logging out using the Log Out option (instead of simply closing your browser) closes your session immediately, freeing up system resources. If you just close your browser, the session remains open on the server until it times out. Web Services uses the Tomcat session timeout. If you want to change the Tomcat session timeout or specify a timeout specific to a Web Services instance, refer to the Tomcat documentation.

To log out of the Admin console

1. Click **Log Out** in the Navigation pane.

The software closes your session and confirms that you have been logged out.

Changing the admin username

You can change the Web Services Admin console username at any time. You must log in to the Admin console to change the username. For more information, see [Logging in to the Admin console on page 42](#).

To change the Admin console username

1. Log in to the Admin console.
2. Click **Change Password** in the Navigation pane.
3. Enter the current console password in the **Current Password** field.
4. Enter a new username in the **New Username** field.
5. Click **Save**.

If the current password is valid, the software saves the new username and returns you to the Status page.

Changing the admin password

You can change the Web Services Admin console password at any time. You must log in to the Admin console to change the password. For more information, see [Logging in to the Admin console on page 42](#).

To change the Admin console password

1. Log in to the Admin console.
2. Click **Change Password** in the Navigation pane.
3. Enter the current Admin console password in the **Current Password** field.
4. Enter a new password in the **New Password** field.
5. Re-type the new password in the **Confirm New Password** field.

Note: Unless you want to create a new admin username, leave the **New Username** field blank.

6. Click **Save**.

If the current password is correct and the new passwords match, the software saves your change and returns you to the Status page.

Viewing the current status of Web Services

The Status page gives you a quick glance at the current status of Web Services. You can see status information about the connection between Web Services and your Symphony ILS, check the version of Web Services, see which clients you have licensed with this instance of Web Services, and check the version of the Symphony ILS that Web Services is connected to. For more information regarding each of the fields on the Status page, see [Fields: Status page on page 46](#).

In addition to the information on the Status page, you can also add additional client licenses (for more information, see [Viewing or updating a Web Services license key on page 45](#)) and reset the Web Services caches (for more information, see [Resetting Web Services caches on page 45](#)).

If you are experiencing connectivity issues between Web Services and your ILS, see [Troubleshooting connection issues on page 84](#) for guidance on how to resolve the issue.

See the following topics for more information:

Viewing or updating a Web Services license key

Each instance of Web Services can have its own license key. License keys are all inclusive; the licenses for every client that you have access to are included in a single file.

Only client IDs that have been added to the license file will be able to access information. Once a license has been added, the included Client IDs need to be assigned to a Symphony interface. Customer name information is included with the license file; this allows licenses to be independently audited.

Clients that are licensed are displayed on the Status page. Any temporary licenses that listed will also have an expiration date associated with the client ID.

To view or update license details

1. Click **Status** in the Navigation pane if the Status page is not already displayed.

The Status page displays the licensed clients in the **Licensed Clients** field.

2. Click **update** at the end of the list of licensed clients.

The **Web Services License** page displays the current licensed client IDs and the license key if installed.

3. To install a license, click **Browse** or **Choose File** (depending on your browser).
4. Select the license file.
5. Click **Save** to save the license and return to the Status page.

If the license file you selected is valid, the new client IDs will display on the Status page. Otherwise, the software will display an error in the message bar.

Resetting Web Services caches

To improve performance, Web Services caches information about some ILS policies and data. Web Services is also able to detect when a connection to Symphony is lost (as the result of a scheduled `auto_haltrun`, for example).

In general, it is not necessary to halt and restart Web Services as a result of a Symphony HIR (halt, initialize, run), even if the halt and restart was required due to ILS configuration changes. In most cases, Web Services will detect a Symphony halt and reset its connections to Symphony.

However, if you note discrepancies in data returned by Web Services after making changes in the ILS (for example, changes to labels or language setup), you can reset the Web Services caches manually from the Status page. Resetting the Web Services caches should resolve any discrepancies that are occurring.

To reset Web Services caches

1. Click **Status** in the Navigation pane.
2. Do one of the following:
 - To reset all Web Services caches except for the security caches, click **Reset Services Only**. For more information about this choice, see [Reset Services Only on page 47](#).
 - To reset all Web Services caches including the security caches, click **Reset All**. For more information about this choice, see [Reset All on page 47](#).

The Web Services caches are cleared and are repopulated as new requests are processed.

Fields: Status page

The Status page displays information about the Web Services connection to the ILS, current software and Web Services versions, and current client licenses. You can also add additional client licenses and reset the ILS connection from this page.

ILS Connection Status

Displays the status of the Web Services connection to the ILS. If the connection is active, it displays **Connected**.

If Web Services cannot connect to the ILS, it displays **Offline**. For guidance on resolving ILS connection issues, see [Troubleshooting connection issues on page 84](#).

Web Services Version

Displays the current version information for this Web Services instance.

Licensed Clients

Displays the Client IDs that this instance of Web Services is licensed for. You can also add additional licenses by clicking **update** at the end of the list. For more information, see [Viewing or updating a Web Services license key on page 45](#).

Note: If you have a temporary license, an expiration date also displays here.

Licensed To

Displays the Customer ID and Customer Name for this group of client licenses.

ILS Version

Displays version information for the SirsiDynix Symphony ILS that this Web Services instance is connected to.

Note: This field will be blank if the ILS Connection Status is **Offline**.

Management Options

Option	Description
Reset All	<p>Lets you clear the following information that Web Services caches during normal operation:</p> <ul style="list-style-type: none"> • Policy information • Other ILS data • Security caches <p>Important: When you clear security caches, all active staff and patron authentication sessions are closed. This requires any staff or patrons to log in again before they can continue to use their applications.</p> <p>For more information, see Resetting Web Services caches on page 45.</p>
Reset Services Only	<p>Lets you clear the following information that Web Services caches during normal operation:</p> <ul style="list-style-type: none"> • Policy information • Other ILS data <p>For more information, see Resetting Web Services caches on page 45.</p>

Updating ILS configuration options

If you make changes to your Symphony ILS configuration or login credentials, you may need to make corresponding changes to the ILS configuration options in the Admin console.

Additionally, making changes to any policy that is used in the Web Services configuration will also require you to update the configuration options in the Admin console.

To update ILS configuration options

1. Choose **ILS Configuration** in the Navigation pane.
2. Update or change the options as needed.

Important: For details about each of the fields you encounter while updating ILS configuration options, see [Fields: ILS Configuration on page 48](#).

3. Choose **Save**.

The software validates the new options by trying to connect to the ILS. This process may take a moment. If the connection is successful, the new configuration is saved.

If the connection fails, verify that you have entered each of the values correctly.

Fields: ILS Configuration

The ILS Configuration page lets you make changes to various Web Services configuration options.

ILS Host

Specifies the IP address or host name of the Symphony ILS server.

This field is required.

netserver Port

Specifies the port number for connecting to the netserver.

This must be a valid port number (between 1 and 65535).

This field is required.

Station User ID

Specifies the SirsiDynix Symphony user to use to establish initial communication with the ILS. This needs to be a Symphony user ID that has administrator-level privileges. This can be an existing user ID or you can create a new user ID specifically for Web Services. For more information about user IDs and User Access in Symphony, refer to the WorkFlows online help.

Note: This value is also used for unprivileged requests, such as searching or checking item availability.

This field is required.

Station Terminal Name

Specifies an ILS station policy. You can review valid values for this property in the Station wizard in the WorkFlows client.

This field is required.

Station Login Clearance

Specifies an Accountability policy. This policy is only used by systems where the Accountability module is enabled. You should leave this as *NONE* (or some other policy that inhibits access). If specific clearance is needed after Web Services starts, it will be determined from the clearance level of the user who is making requests.

This field is required.

Station Login Environment

Specifies an ILS environment policy. Sets the values such as preferred language and limit for sorting search results. You can review the default values for this property in the Environment wizard in the WorkFlows client.

This field is required.

Station Login User Access Code

Sets the privileges for what commands can be executed by Web Services during initialization and for unprivileged requests. This needs to be a User Access policy that has administrator-level privileges. This can be an existing user access policy, or you can create a new user access policy specifically for Web Services. For more information about user IDs and User Access in Symphony, or about working with the User Access Wizard, refer to the WorkFlows online help.

This field is required.

Minimum Sessions

The minimum number of netserver sessions to keep open. You may want to increase this value if your Web Services instance handles a high volume of requests. This value must be a number.

This field is required.

Maximum Sessions

The maximum number of netserver sessions that Web Services will allow to be open. You may want to increase this value if your Web Services instance handles a high volume of requests. This value must be a number.

Important: This value must be greater than or equal to the value of *Minimum Sessions*.

This field is required.

Maximum Idle Time (seconds)

The time (in seconds) an open session to the netserver is allowed to remain idle before it is dropped and is no longer available for use. This value must be a number.

Note: This only affects sessions that are above the number of minimum sessions specified.

This field is required.

Socket Read Timeout (milliseconds)

The maximum amount of time (in milliseconds) that Web Services processes will wait for responses to return from the netserver before timing out. This value must be a number.

This field is required.

Keep Alive Timer (seconds)

How often (in minutes) that Web Services will send keep alive messages to maintain connections between clients. The default value is 0, which turns off keep alive messages. This value must be a number.

For performance reasons, Web Services maintains a pool of open connections to the ILS server. Because some firewalls or network cards close connections that have been idle for some qualified time, setting the **Keep Alive Timer** allows you to force communication across a connection so that the connection will not be dropped.

Consult with your network administrator about any sleep or idle settings in network interfaces in order to determine what time interval may be appropriate for this field.

This field is required.

ILS Timezone

Specifies the time zone of the Symphony instance, which the ILS includes whenever it returns datetime data. The datetime for the client is controlled by Web Services. The time stamp is converted on incoming requests to the Web Services time zone, and the system assumes that outgoing responses are in the same time zone.

Note: The value for your time zone can be any valid Java time zone ID. Commonly, these IDs are a continent followed by a city name, for example, `America/New_York` or `Australia/Sydney`.

Currency Code (Circulation)

Specifies the default currency code that will be used by this instance of Web Services for fines and payment requests. This value can be any valid ISO 4217 currency code.

Always Require Authentication

Requires that all Web Services calls (with a few exceptions such as `login` and `version`) have valid authentication before they will be accepted by Web Services.

By requiring authentication for all calls, any client that makes a request must include a valid `sessionToken`.

Disable Legacy


Specifies that no calls from previous versions of Web Services can be processed while legacy versions are disabled. If you are changing your client code to use only the ROA version, you can use this setting to verify that it does not include any legacy code. Legacy versions include any calls that are under the `rest` or `soap` paths.

Management Options

Option	Description
Save	<p>Saves your changes to the ILS configuration.</p> <p>The software validates the new options by trying to connect to the ILS. This process may take a moment. If the connection is successful, the new configuration is saved.</p> <p>If the connection fails, verify that you have entered each of the values correctly.</p>
Reload	<p>Reloads the current configuration values from the server without saving changes.</p>

Managing Client IDs

Every request processed by Web Services requires a valid client ID (the version request is the only one that does not). Client IDs allow you to limit access to your Web Services instance and your ILS data to only those clients using IDs you specify. Client requests that do not include a valid client ID are rejected.



The client ID received with every request can be logged by Web Services (see [requests log on page 86](#)). This can provide valuable information when troubleshooting or investigating different events that have occurred.

The Client IDs page lists the client IDs that you have specified. These client IDs are the only client IDs that Web Services will accept requests from. You can add as many IDs as you need. We recommend that you provide a unique ID for each client application you want to access this Web Services instance.

You also need to include client IDs for SirsiDynix applications that use Web Services to interact with your ILS (for example, SirsiDynix Enterprise or BookMyne). Refer to the documentation for specific products to learn about the client IDs that those applications use. If you need assistance finding the correct client ID for a SirsiDynix application, contact SirsiDynix Customer Support.

To add, edit, and remove client IDs

1. Click **Client IDs** in the Navigation pane.

The software displays the list of currently configured client IDs.

2. Perform any of the following tasks, as necessary:
 - Edit the name of a client ID from the **Client ID** column.
 - Select a different Symphony Interface for a client ID using the **Symphony Interface** dropdown.
 - Remove a client ID by clicking **Remove** in that ID's row.
 - Add a new client ID by clicking **Add**, then filling in the information for the new client ID.

Important: Client IDs names are case sensitive, so ensure that you have the proper spelling and capitalization for any client IDs that you enter or edit.

Note: For details about each of the fields you encounter while editing client IDs, see **Fields: Client IDs** on page 53.

3. Click **Save** to save your changes.

Fields: Client IDs

Every request processed by Web Services requires a valid client ID (the version request is the only one that does not). Client IDs allow you to limit access to your Web Services instance and your ILS data to only those clients using IDs you specify. Client requests that do not include a valid client ID are rejected.

Client ID

Specifies a unique identifier for a client application.

Important: Client ID values may include only letters, numbers, and underscore characters. Spaces or other punctuation are not allowed.

Symphony Interface

Specifies the type of Symphony interface associated with this client ID. The most commonly used Symphony interfaces are listed below. For a full list of the Symphony interfaces (called client IDs in the WorkFlows interface) that are delivered with a Symphony system, and for instructions about how to add additional Symphony interfaces, see the WorkFlows online help.

Type of client application	Symphony Interface Name	Description
OPAC	Webcat	<p>Specifies an online public access catalog (OPAC) client, such as e-Library.</p> <p>Note: This displays in the WorkFlows client as WEBCAT; the interface ID is 1.</p>
Other Online Catalog	Online Catalog	<p>Specifies a generic online catalog client.</p> <p>Note: This displays in the WorkFlows client as ONLCAT; the interface ID is 5.</p>
Patron Web Services Client	Web Services Public	<p>Specifies a patron Web Services client. A more specific option can be selected for patron clients such as BookMyne (see below). This is the default selection.</p> <p>Note: This displays in the WorkFlows client as WS_PUBLIC; the interface ID is 11.</p>
BookMyne	Web Services BookMyne	<p>Specifies a mobile Web Services client, specifically BookMyne.</p> <p>Note: This displays in the WorkFlows client as BOOKMYNE; the interface ID is 12.</p>

Type of client application	Symphony Interface Name	Description
Discovery	Web Services DS	<p>Specifies either a SirsiDynix Enterprise or SirsiDynix Portfolio discovery client.</p> <p>Note: This displays in the WorkFlows client as WS_DS; the interface ID is 13.</p>
Staff Web Services Client	Web Services Staff	<p>Specifies a staff Web Services client. A more specific option can be selected for staff clients such as BLUEcloud Cataloging (see below).</p> <p>Note: This displays in the WorkFlows client as WS_STAFF; the interface ID is 14.</p>
BLUEcloud PAC	BLUEcloud PAC Client	<p>Specifies a BLUEcloud PAC client.</p> <p>Note: This displays in the WorkFlows client as BC_PAC; the interface ID is 15.</p>
BLUEcloud Cataloging	BLUEcloud Cataloging Staff Client	<p>Specifies a BLUEcloud Cataloging client.</p> <p>Note: This displays in the WorkFlows client as BC_CAT; the interface ID is 16.</p>
BookMyne+	Web Services BookMyne Plus	<p>Specifies a BookMyne+ client.</p> <p>Note: This displays in the WorkFlows client as BOOKMYNE_P; the interface ID is 17.</p>

Type of client application	Symphony Interface Name	Description
Social Library	Social Library Client	Specifies a Social Library client. Note: This displays in the WorkFlows client as SOCIAL_LIB; the interface ID is 18.
MobileCirc	MobileCirc Staff Client	Specifies a MobileCirc client. Note: This displays in the WorkFlows client as MOBLCIRC_S; the interface ID is 19.

Web Services passes the Symphony Interface ID with every request to the ILS. This ID is included in Symphony history logs (dC data code). You can filter logs based on the specified interface.

Symphony Interfaces are also used with user standing maps. Using user standing map policies, you can set up rules to allow or block commands based on a user’s status and the interface a request is coming from.

For more information about history log reports or about user standing maps, refer to the WorkFlows online help.

Management Options

Option	Description
Add	Adds a new client ID to the list of client IDs. After clicking Add , you need to fill in the information for the new client ID and then save your changes.
Remove	Removes a client ID. After removing a client ID, you need to save your changes for the removal to take effect.

Option	Description
Save	Saves the changes you have made to client IDs. Note: If you navigate somewhere else in the Admin console before clicking Save , your changes will be discarded.
Reload	Reloads the current client IDs from the server without saving changes.

Configuring CAS single sign-on

The Single Sign-on Setup page allows you to take advantage of a Central Authentication Service (CAS) server for authentication. From this page you can manage each of your CAS-trusted URLs. For more information about the fields on this page, see [Fields: Single Sign-on Setup on page 57](#).

To configure CAS single sign-on URLs

1. Log in to the Admin console.
2. Click **Single Sign-on Setup** in the Navigation pane.
3. Do one of the following:
 - Click **Add URL** to create a new Single Sign-on URL.
Fill in the **CAS Server URL** and **CAS Service ID**, then click **Save**.
 - Click **Edit** in the row of a Single Sign-on URL that you want to edit.
Make necessary changes and click **Save**.
 - Click **Delete** in the row of a Single Sign-on URL that you want to delete.
The Single Sign-on URL is deleted.

Note: For more details about the fields you encounter while adding or editing a Single Sign-on URL, see [Fields: Create/Edit Single Sign-on URL on page 58](#).

Fields: Single Sign-on Setup

Single Sign-on Setup allows you to manage your trusted CAS URLs.

Single Sign-on server list

Displays a list of all Single Sign-on URLs that may be used with Web Services.

Important: Only the URLs listed in this table may be used for single sign-on authentication.

The list displays this information:

Column	Description
CAS Server	Displays the URL to the CAS server that is running single sign-on services.
CAS Service ID	Displays the URL to the Web Services instance that is using this CAS server.

Management Options

Option	Description
Add URL	Allows you to create a new Single Sign-on URL. For more information see Configuring CAS single sign-on on page 57 .
Edit	Allows you to edit an existing Single Sign-on URL. For more information see Configuring CAS single sign-on on page 57 .
Delete	Allows you to delete a Single Sign-on URL from Web Services. Important: Deleting the URL happens immediately upon clicking Delete ; no warning message displays.

Fields: Create/Edit Single Sign-on URL

In order for single sign-on to function, you must add a valid **CAS Server URL** and **Service ID**.

Property

Displays the CAS Server label.

URL

Specifies the URL for the CAS server that will be used for single sign-on.

Important: The URL must use the SSL/TLS protocol and should be a fully qualified URL.

Important: If this URL will be used for BLUEcloud Central logins, you need to add `/cas` to the end of the URL.

CAS Service ID

Specifies the URL to the Web Services instance that will be using the CAS server. Typically, this is the base URL to this instance of Web Services. For example, *http://testlibrary.sirsidynix.com/symws*.

Version

Specifies the CAS version to use for logins. Typically, you will use the CAS version that is supported by your server.

Note: If you need to maintain compatibility with the `user/patron/login` and `security/loginUserCAS` actions, use CAS 1.0.

Note: If you are connecting to BLUEcloud Central, you can use either CAS 2.0 or CAS 3.0, depending on your server.

Type

Specifies the user type that will be using this URL for logins. Specifying the type can speed up login by excluding different types of SSO logins. You can use Both to allow all types of SSO logins to be included.

Management Options

Option	Description
Save	Saves the new URL (creating) or saves changes to the existing URL (editing).
Cancel	Returns to the previous page without saving any changes.

Related topics

[Configuring CAS single sign-on on page 57](#)

[Fields: Single Sign-on Setup on page 57](#)

LDAP Setup

Web Services allows you to specify one or more LDAP servers to authenticate users as they make Web Services client requests.

See the following topics for more information:

About LDAP Authentication

The Lightweight Directory Access Protocol (LDAP) provides the ability to search and manage data in a network-accessible directory service. An LDAP service is commonly used to store data about users, such as the user's name, contact information, department, and so on. It can also be used as a common source for user authentication (for example, allowing multiple applications to validate login credentials, such as a user ID and password).

If your library already uses LDAP authentication for your OPAC, you will likely want to configure the same LDAP servers for use with Web Services.

In order to use LDAP authentication in Web Services, you must supply a Web Auth ID as part of the patron record in the ILS for each library user that will authenticate using LDAP. Contact SirsiDynix Customer Support if you need help configuring or importing Web Auth IDs.

The Web Auth ID for a patron record must be the same as the value of the *Search Attribute* (for more information, see [Fields: Add/Edit LDAP Server on page 64](#)) for that user in the LDAP directory (using either the same case or all uppercase). Understanding how Web Services uses LDAP will help to illustrate.

Example of a Web Services and LDAP exchange

Web Services login requests require a login token (a user ID, for example) and a password for a user. Say, for example, that a client needed to authenticate Sally Smith whose user ID is *ssmith*.

If you configure LDAP servers in the Web Services Admin, when Web Services receives a request to authenticate or log in a user, it will attempt to contact the first LDAP server in the list. It first searches for the user using the Search Attribute you specify for the server and the login token in the request, for example *uid=ssmith*.



If the LDAP server does not allow anonymous searching, Web Services attempts to bind (log in) to the server before searching.

If it finds an entry for *ssmith*, it uses the distinguished name for that LDAP entry and the password supplied in the request to attempt to bind to the directory as that user.

If the bind succeeds, Web Services then uses the value of the *Search Attribute* (in this case, the value of the *uid* attribute in the directory response) to look for a user in the ILS with *ssmith* as the Web Auth ID.

Case is significant in this exchange. If the LDAP directory reports the value as *SSmith*, the Web Services software searches for a patron record with the Web Auth ID of *SSmith*. If no match is found, Web Services tries to match a Web Auth ID of *SSMITH*. If the Web Auth ID for the patron record is entered as *ssmith* (all lowercase), it will not match.

If the user is not found in the first LDAP directory, Web Services tries the next LDAP server and so on until it succeeds or until it has tried all LDAP servers.

If Web Services is unable to connect to an LDAP server, it will log an error in the `symws` log (see [Examining log files on page 86](#)).

By default, if authentication fails for all LDAP servers, Web Services will attempt to authenticate the user with the ILS. This allows you to authenticate users who may exist only in the ILS and not on LDAP servers. If you want Web Services to use only LDAP authentication, enable the LDAP Only Authentication option (for more information, see [Fields: LDAP Setup on page 63](#)).



If LDAP is taking too long to fail the authentication, you can change the timeout values (see [Changing the LDAP Timeout Settings on page 62](#)).

Managing LDAP Server Connections

The LDAP Setup page lets you specify and manage LDAP servers used by Web Services for user authentication and login.

Each new server you add to the list appears at the top. The order of servers in the list determines the order that Web Services will use as it authenticates users. That is, when a request is made to authenticate or log in a user, Web Services will contact the first LDAP server in the list. If authentication fails, it will try the next server in the list, and so on until it successfully authenticates or has tried all servers.

By default, if authentication fails for all LDAP servers, Web Services will attempt to authenticate the user with the ILS. If you want Web Services to use only LDAP authentication, enable the LDAP Only Authentication option.

For more information about the fields on this page, see [Fields: LDAP Setup on page 63](#).

To add, edit, and remove LDAP Server Connections

1. Log in to the Admin console.
2. Click **LDAP Setup** in the Navigation pane.

The software displays the list of currently configured LDAP servers.

3. Do one of the following:
 - Click **Add LDAP Server** to create a new LDAP server.
Fill in the required fields, and then click **Save**.
 - Click **Edit** in the row of a LDAP Server that you want to edit.
Make necessary changes and click **Save**.
 - Click **Delete** in the row of a LDAP server that you want to delete.

The LDAP server is deleted.

For more details about the fields you encounter while adding or editing a Single Sign-on URL, see [Fields: Add/Edit LDAP Server on page 64](#).

Changing the LDAP Timeout Settings

If login requests through LDAP take too long to fail when the user is not matched in the LDAP repository, you can change the timeout value.

To change the LDAP timeout settings

1. Go to the directory where Tomcat is installed on your system.
2. Navigate to the `webapps/<application name>/WEB_INF/classes` directory.
3. Open the `ldapConfiguration.properties` file in a text editor.
4. Locate the `ldap.connectTimeout` and `ldap.readTimeout` values for the appropriate LDAP server.

Note: The values are numbered to correspond with the LDAP server entry.

5. Change the values to numbers lower than the defaults.

Note: The default for `ldap.connectTimeout` is 500 milliseconds.

Note: The default for `ldap.readTimeout` is 5000 milliseconds.

6. Save the file.

Fields: LDAP Setup

The LDAP Setup page lets you specify and manage LDAP servers used by Web Services for user authentication and login.

Each new server you add to the list appears at the top. The order of servers in the list determines the order that Web Services will use as it authenticates users. That is, when a request is made to authenticate or log in a user, Web Services will contact the first LDAP server in the list. If authentication fails, it will try the next server in the list, and so on until it successfully authenticates or has tried all servers.

By default, if authentication fails for all LDAP servers, Web Services will attempt to authenticate the user with the ILS. If you want Web Services to use only LDAP authentication, enable the LDAP Only Authentication option.

LDAP server list

Displays a list of all the LDAP servers that Web Services will use to authenticate users.

The list displays this information:

Column	Description
LDAP Host	Displays the host name or IP address for the listed server.
Base Distinguished Name	Displays the base distinguished name specified for the listed server.

Management Options

Option	Description
LDAP Only Authentication	<p>Specifies how Web Services should handle user authentication. Click disable or enable to toggle this setting.</p> <p>disabled: If LDAP authentication fails, attempt to authenticate with the ILS server. Use this option if you have some users that validate through LDAP and others that are not part of LDAP directories.</p> <p>enabled: Only authenticate with LDAP (more secure).</p> <p>Note: This option only appears when there are servers in the list.</p>
Add LDAP Server	<p>Allows you to create a new LDAP server. For more information see Managing LDAP Server Connections on page 61.</p>
Edit	<p>Lets you edit the selected server.</p>
Delete	<p>Deletes the selected server configuration and removes it from the list.</p>

Fields: Add/Edit LDAP Server

Add/Edit LDAP Server lets you configure connection properties for an LDAP server.

LDAP Host

Specifies the host name or IP address of the LDAP server. For example, *ldap.example.org* or *10.1.1.116*.

This field is required.

Port

The port number for the LDAP service on the server. The default LDAP port is 389, or 636 for LDAP over SSL. If you are using Microsoft Active Directory, this would typically be the port for the global catalog. This must be a valid port number (between 1 and 65535).

This field is required.

SSL Authentication

Specifies whether Web Services should attempt to connect to the LDAP server using the *ldaps:* scheme. Select Yes if the LDAP directory requires SSL. The default value is No.

Note: By default, Web Services accepts all security certificates. If you want to validate certificates, you will need to edit the `applicationContext.xml` file in the WEB-INF directory and change `acceptAllCerts` to false. You must also import certificates using a tool such as *keytool* or *javacpl*.

For more information on client and server authentication, refer to the documentation for Apache Tomcat and Java. Managing a Tomcat trust store is beyond the scope of this document.

Base Distinguished Name

Specifies the base distinguished name for entries in the LDAP directory. For example, for the user `uid=myuser,ou=student,dc=ldap,dc=example,dc=org`, the base distinguished name would be `ou=student,dc=ldap,dc=example,dc=org`.

Web Services will do a recursive tree search from this level of the directory hierarchy, including any referrals returned by the directory.

This field is required.

Search Attribute

Specifies the unique attribute that distinguishes an entry in the directory. Commonly, this is the *uid* attribute.

For example, in a directory with unique entries such as `uid=ssmith,ou=student,dc=ldap,dc=example,dc=org`, *uid* would be the value to use for Search Attribute.

This field is required.

Distinguished Name for BIND

Specifies the full distinguished name for a named entry in the directory. For example, `uid=searchuser,ou=admin,dc=ldap,dc=example,dc=org`.

If your LDAP directory requires BIND (that is, it requires a client to authenticate before searching and it does not allow anonymous searching), you must specify a valid distinguished name for bind and provide the password for that entry in BIND Password.

This field is optional. Leave this field blank if the LDAP directory allows anonymous searching.

BIND Password

Specifies the password for the named entry you entered in **Distinguished Name for BIND**.

This field is optional. Leave this field blank if the LDAP directory allows anonymous searching.

Confirm BIND Password

Confirms that the **BIND Password** was entered correctly. Retype the BIND password.

Note: If the values of **BIND Password** and **Confirm BIND Password** don't match exactly, the LDAP server will not be saved.

This field is required if you have typed a password into **BIND Password**.

Management Options

Option	Description
Save	Saves the changes you have made to client IDs. Note: If you navigate somewhere else in the Admin console before clicking Save , your changes will be discarded.
Cancel	Closes the Add/Edit LDAP Server page without saving the changes and returns you to LDAP Setup.

Related topics

[About LDAP Authentication on page 60](#)

[Fields: LDAP Setup on page 63](#)

[LDAP Setup on page 60](#)



Patron Primary Address Configuration

The Patron Primary Address Configuration page allows you to map client address fields to SirsiDynix Symphony simple entry IDs. This mapping ensures that address information conveyed from the client is formed using the address format your SirsiDynix Symphony system uses and stored in the correct address fields in the SirsiDynix Symphony database.

You can configure all three addresses available in the user record (Address 1, Address 2, and Address 3). To map an address field using a combination of two or more entry IDs, insert a plus sign character [+] between each entry ID in the field (for example, CITY+STATE). Web Services inserts a space between each value in the interface.



You can only configure preferred addresses using simple entry IDs; using entries from synonym lists is not supported.

Follow these instructions to configure patron primary addresses.

To configure patron primary addresses

1. Log in to the admin console.
2. Choose **Patron Primary Address Configuration**.
3. Enter the entry IDs appropriate for your SirsiDynix Symphony system's address configurations into the address fields. For more information, see [Fields: Patron Primary Address Configuration on page 67](#).
4. Choose **Save** to save your changes.

Fields: Patron Primary Address Configuration

The Patron Primary Address Configuration page allows you to map client address fields to SirsiDynix Symphony simple entry IDs. You can configure all three addresses available in the user record (Address 1, Address 2, and Address 3).

To map an address field using a combination of two or more entry IDs, insert a plus sign character [+] between each entry ID in the field (for example, CITY+STATE). Web Services inserts a space between each value in the interface.



You can only configure primary addresses using simple entry IDs; using entries from synonym lists is not supported.

Area

Specifies the entry ID (or entry IDs) that defines the address's area.

Postal Code

Specifies the entry ID (or entry IDs) that defines the address's postal code.

Country

Specifies the entry ID (or entry IDs) that defines the address's country.

Line 1

Specifies the entry ID (or entry IDs) that defines the first line of the address.

Line 2

Specifies the entry ID (or entry IDs) that defines the second line of the address.

Line 3

Specifies the entry ID (or entry IDs) that defines the third line of the address.

Line 4

Specifies the entry ID (or entry IDs) that defines the fourth line of the address.

Line 5

Specifies the entry ID (or entry IDs) that defines the fifth line of the address.

Email Address

Specifies the entry ID (or entry IDs) that defines the email address associated with the address.

Phone Number

Specifies the entry ID (or entry IDs) that defines the phone number associated with the address.

Resource Security Configuration

Web Services allows you to specify which fields patron users can view or modify within their own account. This configuration also allows you to disable access to resources that you do not want patron users to view or modify.

See the following topics for more information:

About the Resource Security Configuration	69
Setting resource security	69
Fields: /circulation/block	70



Fields: /circulation/block/detailLine	71
Fields: /user/patron	72
Fields: /user/patron/address1	76
Fields: /user/patron/address2	76
Fields: /user/patron/address3	76

About the Resource Security Configuration

Within the Resource Security Configuration, you can set the security of fields within the following resources:

- /circulation/block
- /circulation/block/detailLine
- /user/patron
- /user/patron/address1
- /user/patron/address2
- /user/patron/address3

You can set fields to be editable by patron users, set fields to be visible but not editable, or disable visibility of fields altogether. Within each resource, you can set the security for each available field individually.

The three security options that may be available are the following:

- Disabled – If a field is specified as Disabled, patron users will not be able to see or modify the field.
- Read-only – If a field is specified as Read-only, patron users will be able to see the field's value, but will not be able to modify it.
- Editable – If a field is specified as Editable, patron users will be able to see and modify the field's value.



Some fields do not have the option to be set as Editable.

For information on how to set these fields, see [Setting resource security on page 69](#).

Setting resource security

To set the security for resource fields, do the following.

To configure resource security access

1. Log in to the Admin console.
2. Click **Resource Security Configuration** in the Navigation pane.
3. Click the name of the resource that has the fields you want to configure.

The Resource Security Page Detail for that resource displays.

4. Set the security settings for each field as appropriate to your site's needs. For more information on these fields, see one of the following topics:
 - [Fields: /circulation/block on page 70](#)
 - [Fields: /circulation/block/detailLine on page 71](#)
 - [Fields: /user/patron on page 72](#)
 - [Fields: /user/patron/address1 on page 76](#)
 - [Fields: /user/patron/address2 on page 76](#)
 - [Fields: /user/patron/address3 on page 76](#)
5. Click **Save** to save your changes, or click **Cancel** to clear your changes.

Fields: /circulation/block

The /circulation/block page allows you to specify the security level of the fields within the /circulation/block resource.

Attributes

Set the following fields to Hidden, Read-only, or Editable; for more information about these values, see [About the Resource Security Configuration on page 69](#).

Field	Description
amount	Specifies the money amount of the bill.
billDate	Specifies the date a bill was created.
block	Specifies a block on the patron user's record.

Field	Description
comment	Specifies a comment to be associated with the patron user's block.
createDate	Specifies the date the block was created.
detailLineList	Specifies the list of details associated with blocks on the patron user's record.
estimatedOverdueAmount	Specifies the estimated fee for an overdue item on the patron user's record.
item	Specifies an item associated with the patron user's record.
library	Specifies the library to which the patron user belongs.
owed	Specifies the money amount the patron user owes in fines and fees to the library.
patron	Specifies the patron user associated with the block.

Fields: /circulation/block/detailLine

The /circulation/block/detailLine page allows you to specify the security level of the fields within the /circulation/block/detailLine resource.

Attributes

Set the following fields to Hidden, Read-only, or Editable; for more information about these values, see [About the Resource Security Configuration on page 69](#).

Field	Description
amount	Specifies the amount the patron user's paid to clear the block.

Field	Description
createDate	Specifies the date the block was created.
paymentLibrary	Specifies the library at which the patron user made the payments to clear the block.
paymentType	Specifies the type of payment the patron user used to pay the fine or fee that created the block.

Fields: /user/patron

The /user/patron page allows you to specify the security level of the fields within the /user/patron resource.

Attributes

Set the following fields to Hidden, Read-only, or Editable; for more information about these values, see [About the Resource Security Configuration on page 69](#).

Field	Description
access	Specifies the access level of the patron user.
address1	Specifies an address that can be used for the patron user.
address2	Specifies an address that can be used for the patron user.
address3	Specifies an address that can be used for the patron user.
alternateID	Specifies the alternate ID of the patron user.
barcode	Specifies the barcode associated with the patron user.
birthDate	Specifies the date of birth of the patron user.

Field	Description
blockList	Specifies the list of blocks on the patron user's record.
category1	Specifies a patron user statistic value.
category2	Specifies a patron user statistic value.
category3	Specifies a patron user statistic value.
category4	Specifies a patron user statistic value.
category5	Specifies a patron user statistic value.
category6	Specifies a patron user statistic value.
category7	Specifies a patron user statistic value.
category8	Specifies a patron user statistic value.
category9	Specifies a patron user statistic value.
category10	Specifies a patron user statistic value.
category11	Specifies a patron user statistic value.
category12	Specifies a patron user statistic value.
checkoutLocation	Specifies the library at which the patron user checks out items, if New Profile feature for SirsiDynix Symphony is enabled.

Field	Description
circHistoryRecordList	Specifies the display of the patron user's circulation history.
circRecordList	Specifies the list of the patron user's checked out items.
createdDate	Specifies the date the patron user's record was created.
customInformation	Specifies any extended information associated with the patron user's record.
department	Specifies the value set for the Department field in this patron user's record.
displayName	Specifies how the patron user wants his or her name to display in library communications.
environment	Specifies the environment to which the patron user belongs.
estimatedOverdueAmount	Specifies the estimated fee for an overdue item on the patron user's record.
firstName	Specifies the first name of the patron user.
holdRecordList	Specifies the list of holds the patron user currently has on his or her record.
language	Specifies the language the patron user wants to use for library communications.
lastActivityDate	Specifies the last date the patron user used his or her library account.
lastName	Specifies the last name of the patron user.

Field	Description
library	Specifies the library to which the patron user belongs.
middleName	Specifies the middle name of the patron user.
pin	Specifies the PIN of the patron user.
preferredAddress	Specifies the address at which the patron user prefers the library to use.
preferredName	Specifies the name which the patron user prefers to have listed in library communication.
privilegeExpiresDate	Specifies the date on which the patron user's circulation privileges expire.
privilegeGrantedDate	Specifies the date on which the patron user's circulation privileges were granted.
profile	Specifies the user profile of the patron user.
standing	Specifies the patron user's standing.
suffix	Specifies the suffix of the patron user's name, if applicable.
title	Specifies the title of the patron user, if applicable.
usePreferredName	Specifies whether or not the library system should use the patron user's preferred name in library communications.
webAuthID	Specifies the Web Auth ID of the patron user.

Fields: /user/patron/address1

The /user/patron/address1 page allows you to specify the security level of the fields within the /user/patron/address1 resource.

Attributes

Set the following fields to Hidden, Read-only, or Editable; for more information about these values, see [About the Resource Security Configuration on page 69](#).

Field	Description
code	Specifies the type of field the address contains. For example, STREET, PHONE, or ZIP).
data	Specifies the value that corresponds to the specified code.

Fields: /user/patron/address2

The /user/patron/address2 page allows you to specify the security level of the fields within the /user/patron/address2 resource.

Attributes

Set the following fields to Hidden, Read-only, or Editable; for more information about these values, see [About the Resource Security Configuration on page 69](#).

Field	Description
code	Specifies the type of field the address contains. For example, STREET, PHONE, or ZIP).
data	Specifies the value that corresponds to the specified code.

Fields: /user/patron/address3

The /user/patron/address3 page allows you to specify the security level of the fields within the /user/patron/address3 resource.

Attributes

Set the following fields to Hidden, Read-only, or Editable; for more information about these values, see [About the Resource Security Configuration on page 69](#).

Field	Description
code	Specifies the type of field the address contains. For example, STREET, PHONE, or ZIP).
data	Specifies the value that corresponds to the specified code.

Managing the log files

Web Services for Symphony keeps logs on a variety of interactions within Web Services. You can use these log files to locate and fix any bugs that you might be experiencing in your implementation. You can also enable a debug mode for specific loggers to display more data that can help you isolate issues in the code.

This section includes these topics:

Viewing Web Services Logs	77
Changing the logging level	78
Fields: Logs	78

Viewing Web Services Logs

The Admin console provides the capability to access and view the log files for Web Services.

To view Tomcat logs

1. Log in to the Admin console.
2. Choose **Logs** in the Navigation pane.

A list of all the logs for your Web Services instance displays. The system looks in the Tomcat logs directory for logs that start with your Web Services instance name and end with the .log extension.

Important: If the Logs option does not display, the path to the default log directory is incorrect and needs to be updated. To update the path, change the value for the `logging.directory` property in the `admin-settings.properties` file of the Web Services instance, for example:

```
<tomcat\webapps\symws>\WEB-INF\classes\admin-settings.properties
```

where `<tomcat\webapps\symws>` is a path to your instance of Web Services.

3. Choose a log name from the list to open or download the log file to your computer.
Note: Although the requests log may appear in the logs list, it may be empty if you have not enabled request logging. For more information, see [requests log on page 86](#).

Related topics

[Managing the log files on page 77](#)

[Changing the logging level on page 78](#)

Changing the logging level

You can toggle between the INFO and DEBUG log levels. The Log Level column displays the current logging status in the Apache Tomcat server.

1. Log in to the Admin console.
2. Choose **Logs** in the Navigation pane.

A table of all the loggers for your Web Services instance displays below the list of log files.

3. Choose **Change Level** to toggle the level of the current logging level.

Note: Choosing Change Level switches the logging level from the current level to DEBUG. When the level is DEBUG, Change Level sets the logging level to INFO. From Web Services, you cannot change the logging level to OFF or any other level besides DEBUG or INFO. To change logging to a different level other than DEBUG or INFO, you must change it from within Apache Tomcat.

Related topics

[Managing the log files on page 77](#)

[Viewing Web Services Logs on page 77](#)

[Fields: Logs on page 78](#)

Fields: Logs

The Logs page lets set the logging level of specific loggers and to download log files to review.

Log file list

Displays a link to download the log file from the server.

Log status table

Lets you select the level of logging for specific log files.

Option	Description
Log Name	Displays the logger that creates the log file.
Log Level	The mode that the logger creates the log at. The options are either OFF (normal) or DEBUG (full message logging).

Related topics

[Viewing Web Services Logs on page 77](#)

Customizing or localizing labels and messages

You can create a list of labels to replace those that Web Services for Horizon and Symphony uses by default. Your changes are retained through any upgrades to Web Services that may follow. In addition, you can create translations for labels that are used whenever Web Services responds to requests that include that language.

When sending a request, include the language code in the `x-sirs-locale: en-US` header.

To create a customized labels properties file

- Do one of the following:
 - To customize labels in the Web Services Admin Console, in the WEB-INF/classes directory, create a text file named `adminResource_lg_lc_custom.properties` where *lg_lc* is the language/locale code for the labels.
 - To customize exception messages, in the WEB-INF/classes directory, create a text file named `IlsWsExceptionResource_lg_lc_custom.properties` where *lg_lc* is the language/locale code for the labels.
- Locate the resource file for the customized file you are creating:
 - The administration labels are located in WEB-INF/classes/adminResources.properties.

- The exception messages are located the `IlsWsExceptionResource.properties` file in `WEB-INF/lib/ilsws-util.jar`. (You will need an archive extraction tool, such as 7-Zip to open the file.)

Important: You will not edit these files. Any changes made to these files will be overwritten the next time you upgrade Web Services.

3. Open the resource file and locate the string you want to customize.

Each line consists of a key/value pair, as shown in this example:

```
unableToLogin=Unable to log in.
```

4. Copy the key portion ("unableToLogin" in the example) of the line.
5. Paste the key into the `adminResource_Lg_Lc_custom.properties` or `IlsWsExceptionResource_Lg_Lc_custom.properties` that you created in step 1.
6. Type "=" and the string value for the customized message or label.

For example:

```
unableToLogin=Impossibile fare il log in.
```

7. When you have finished, save the file.
8. Restart Tomcat to activate the changed labels or messages.

Uninstalling Web Services

When you install Web Services, the installer creates a directory named `_Web Services` for `Symphony_installation` in the target Tomcat directory. That directory includes an executable named `Uninstall Web Services for Symphony`.

To uninstall Web Services, simply launch the uninstaller executable. Any files you added or modified after installation will not be removed. The uninstaller will list files and directories that could not be removed so that you may remove them manually if you desire.



You may need to run the uninstaller as a system administrator user (Windows) or as the root user (Linux or Solaris).



When installing on Windows, the installer also registers the uninstaller so that you can run it using Add or Remove Programs.

Troubleshooting

This section provides information about where to look for logs to detect problems you may encounter while running Web Services. It also provides troubleshooting information about common errors that you may encounter while using Web Services.

See the following topics for more information:

General Troubleshooting	82
Restoring access if you cannot log in	83
Starting up Tomcat	83
Troubleshooting connection issues	84
Verifying that Web Services is running	85
Examining log files	86
catalina logs	86
requests log	86
symws log	87
BlackBox log	88
Troubleshooting common errors	89
BeanCreationException: Error creating bean	89
BindException: Address already in use: JVM_Bind	89
Connection refused - Cannot connect to host	90
Context initialization failed	90
HATIOException	90
HatIOException: Read of response timed out	91
HatSessionException: Unable to acquire session with server	91
Initializing session manager failed	92
Initial LDAP bind failed	92
listenerStart error	93

General Troubleshooting

This section describes issues that you can encounter while using Web Services and gives instructions about actions you can take in each scenario to try to resolve the problem.

See the following topics for more information:

Restoring access if you cannot log in

The username and password for the Admin console are stored in a file called `admin-settings.properties` in the Web Services instance, for example:

```
<tomcat base path\webapps\symws>\WEB-INF\classes\admin-settings.properties
```

where `<tomcat base path\webapps\symws>` is the path to your Web Services instance.

If you forget the username, you can simply look in this file to remind you. The password, however, is encrypted.

If you forget the password, you can delete the password in this file, save the file, and restart Tomcat.

When you next log in to the Admin console, the Admin console detects that the password is missing. The Admin console will allow you to log in using the default password, `admin`. Be sure to change the password once you log in.

Starting up Tomcat

After you install Web Services you need to start the Tomcat service (Windows only) or Tomcat instance (Windows, Linux, or Solaris) before Web Services can start receiving requests from client applications. After starting Web Services, you can verify that they are running by using a couple of methods. If you did not install Tomcat as a Windows service, then you should follow the Windows instructions for starting up the Tomcat instance. For more information, see [Verifying that Web Services is running on page 85](#).

Starting up the Tomcat Service (Windows only)

1. Open the **Local Services** window.
2. Locate *Apache Tomcat* `<tomcat service name>` in the lists of services, where `<tomcat service name>` is the name you specified for the Tomcat service when you installed Web Services.
3. Right-click the service in the row and select **Start**.

You can look at the status of a service in the **Status** column to verify that it is running. Services which are currently running display a status of **Started**.

Starting up the Tomcat instance (Windows, Linux, or Solaris)

1. Ensure that you have changed your Java environment variable. For more information, see [Finding and setting Java environment variables on page 18](#). (Linux

- or Solaris)
2. Open a command prompt (Windows) or terminal window (Linux or Solaris).
 3. Navigate to the Tomcat root directory for this instance of Web Services.
 4. Run the `startup.bat` script (Windows) or `startup.sh` script (Linux or Solaris) to start the Tomcat instance.

Troubleshooting connection issues

The Admin console Status page reports the status of the Web Services connection to the ILS. If for some reason Web Services cannot connect, it will report “Offline”. For more information see [Fields: Status page on page 46](#).

Here are some reasons the connection may be offline:

Cause	Remedy
The netserver or ILS is not running.	Verify that the ILS and the netserver are running.
The netserver or ILS is inaccessible.	Verify that there are no firewalls blocking access. Inspect for port conflicts.
The hostname or IP address of the netserver may be incorrect.	Verify the hostname or IP address on the ILS Configuration page. For more information, see Fields: ILS Configuration on page 48 .
There is a problem resolving the hostname.	Verify that DNS is correctly resolving the hostname by using <code>nslookup</code> or a similar tool. For more information, see the documentation for your operating system.
The netserver port is incorrect.	Verify the port number on the ILS Configuration page. For more information, see Fields: ILS Configuration on page 48 .
There is a problem with the station login information.	Verify that valid values are used on the ILS Configuration page for Station User ID, Station Terminal Name, Station Login Clearance, Station Login Environment, and Station Login User Access Code. For more information, see Fields: ILS Configuration on page 48 .

For additional tips and information about specific errors, see [Troubleshooting common errors on page 89](#).

Verifying that Web Services is running

After you install Web Services and start Tomcat, you can verify that Web Services is running by pointing a Web browser to the Web Services Admin console (for more information, see [Accessing the Admin console on page 40](#)) and logging in (for more information, see [Logging in to the Admin console on page 42](#)).

If Web Services is running and properly configured, ILS Connection Status will display “Connected” and you will see version information for Web Services and for the ILS.

If you do not see the expected result, see [Troubleshooting common errors on page 89](#) for help with any errors that may have been logged.

Alternately, you can verify that Web Services is running by using a Web browser to send a version request to the URL for your service instance.

For example, if your Tomcat host name is *libraryapps.example.org* and you used the default port (8080) and the default application name (`symws`), the URL for a version request would be:

```
http://libraryapps.example.org:8080/symws/rest/standard/version
```

If Web Services is running and properly configured, the software will return an XML document with version information about Symphony and Web Services, for example:

```
<VersionResponse xmlns="http://schemas.sirsiidynix.com/symws/standard">
  <version>
    <product>SYMPHONY</product>
    <version>3.4.1.5.GA</version>
  </version>
  <version>
    <product>SYM-WS</product>
    <version>v5_0_4879M 2014-10-14 16:16:10</version>
  </version>
</VersionResponse>
```

If you do not see the expected result, see [Troubleshooting common errors on page 89](#) for help with any errors that may have been logged.

If you see version information for SYM-WS (Web Services) but not Symphony, there is a problem with the connection to the netserver. For more information, see [Troubleshooting connection issues on page 84](#).

If network cards or firewalls in your network are configured to close connections that have been idle for some qualified time, you may also want to set the Keep Alive Timer after installation. For more information, see [Fields: ILS Configuration on page 48](#).

Examining log files

Errors encountered while starting and running Web Services for Symphony are recorded in log files. Some of these log files are used only by Tomcat, while others are used only by Web Services for Symphony. These logs can be helpful in detecting and diagnosing problems. This section describes where to find the different types of log files, and details the information that you will find in each file. For more information about specific errors, see [Troubleshooting common errors on page 89](#).

See the following topics for more information:

catalina logs

Errors encountered while starting Web Services are logged in the standard catalina log file. By default, catalina logs are created in the logs directory of your Tomcat installation, for example:

```
<Tomcat directory path>/logs/catalina.2018-11-04.log
```

In some cases, errors may also be logged to the localhost log in the Tomcat logs directory.



Because catalina logs are created and used only by Tomcat, please see the Tomcat documentation for more information about messages that occur in the catalina log files.

requests log

Each request received by Web Services may be logged to a requests log in the Tomcat logs directory.

The requests log uses the name you specified for your Web Services instance. For example, if you used the default application name, *symws*, when you installed Web Services, the requests log would be

```
<Tomcat directory path>/logs/symws-requests.log
```

If instead you used, for example, *mylibraryws* as the application name, the *symws* log file would be

<Tomcat directory path>/logs/mylibraryws-requests.log

The format of entries in this log may change in future revisions.

Requests logging is disabled by default. To enable requests logging, you must modify `logback.xml` in the Web Services classes directory.



Changes you make to `logback.xml` are not persistent across upgrades. If you enable request logging and later upgrade your instance of Web Services, you will need to restore the changes after upgrade.

To turn on request logging in `logback.xml`

1. Open `<tomcat/webapps/symws>/WEB-INF/classes/logback.xml` in a text editor (where `<tomcat/webapps/symws>` is the path to your Web Services instance).
2. Locate the following logger elements:

```
<!-- to enable client request logging change OFF settings to INFO -->
<logger
  name="com.sirsidynix.symws.protocol.rest.handlers.HeaderHandler"
  additivity="false">
  <level value="OFF"/>
  <appender-ref ref="REQUEST"/>
</logger>
<logger
  name="com.sirsidynix.symws.protocol.soap.handlers.HeaderHandler"
  additivity="false">
  <level value="OFF"/>
  <appender-ref ref="REQUEST"/>
</logger>
<logger name="com.sirsidynix.ilsws.license.WSLicenseFilter"
  additivity="false">
  <level value="OFF"/>
  <appender-ref ref="REQUEST"/>
</logger>
```

3. Change the level value from *OFF* to *INFO*:

```
<level value="INFO"/>
```

4. Save the changes to the `logback.xml` file.
5. Restart Tomcat.

symws log

Web Services runtime errors and information messages are logged in the `symws` log in the Tomcat logs directory.

The symws log uses the name you specified for your Web Services instance. For example, if you used the default application name, *symws*, when you installed Web Services, the symws log would be

```
<Tomcat directory path>/logs/symws.log
```

If instead you used, for example, *mylibraryws* as the application name, the symws log file would be

```
<Tomcat directory path>/logs/mylibraryws.log
```

You can access these log files through the Admin console. For more information, see [Viewing Web Services Logs on page 77](#).

BlackBox log

Web Services for Symphony includes a logger for viewing all of the JSON requests and responses that are made in the system. The `blackbox.log` file logs these transactions. You can turn on the BlackBox transaction logger from within the `logback.xml` file.



Because the BlackBox logger creates a highly detailed log, you can use it to troubleshoot specific calls in a controlled environment over a limited time span. However, if you leave the BlackBox logger activated for an extended time period, the log files could easily grow to fill your server's hard drive.

To activate the BlackBox logger

1. Open the `logback.xml` file in the `classes` directory.
2. Locate the following "BlackBox" logger element in the file.

```
<logger name="BlackBox" level="OFF" additivity="false">
  <appender-ref ref="blackbox"/>
</logger>
```

3. Change the level value from *OFF* to *DEBUG*.
4. Save the file.
5. Either wait about a minute for Tomcat to process the edited file or restart Tomcat.

After Tomcat has processed the file, the logging begins. As Web Services for Symphony runs requests, they are added to the `blackbox.log` file.

Note: The BlackBox logger does not log legacy web services transactions.

6. When you have finished logging transactions, disable logging by returning the logging level to *OFF* in the `logback.xml` file and saving it. The logging stops after Tomcat has processed the changes to the file.

Troubleshooting common errors

This section describes some of the error messages you may encounter while using Web Services. Errors are recorded in various Web Services logs (see [Examining log files on page 86](#)).

See the following topics for more information:

BeanCreationException: Error creating bean

This startup error indicates a Java incompatibility or configuration error.

Possible Causes

- You are using a Java version earlier than 1.8u221.
- Required configuration values, such as host or port, are missing.

Solutions

- Make sure you are using the correct Java version. For more information, see [Java software on page 4](#).
- Verify that the host name or IP address and port for the netserver are correct. For more information, see [Fields: ILS Configuration on page 48](#).

BindException: Address already in use: JVM_Bind

This startup error indicates that a port needed by Tomcat is already in use.

Possible Causes

- One of the ports specified in the Tomcat `server.xml` configuration file (in the Tomcat conf directory) is already in use by another application or by another instance of Tomcat already running.

Solutions

- If the port conflict is for HTTP (default 8080) or HTTPS (default 8443), you can reinstall and specify different ports that are not used, or you can modify the `server.xml` directly.

- If the port conflict is for server shutdown (default 8005) or AJP 1.3 (default 8009), you must modify `server.xml` to specify ports that are not used.

Refer to the Tomcat documentation for information about `server.xml`. Refer to your operating system documentation for information about troubleshooting port conflicts, including the use of the `netstat` command to view ports in use.

Connection refused - Cannot connect to host

This startup error indicates a problem with the connection to the ILS.

Possible Causes

- The ILS server or netserver is not available.
- The connection information for the netserver host or port is invalid.

Solutions

- Verify that the host name or IP address and port for the netserver are correct. For more information, see [Fields: ILS Configuration on page 48](#).
- Make sure the netserver and ILS are running.

Context initialization failed

This startup error typically indicates a Java incompatibility.

Possible Causes

- This error is commonly caused by using an earlier version of Java (older than 1.8u221).

Solutions

- Make sure you are using the correct Java version. For more information, see [Java software on page 4](#).

HATIOException

This runtime exception indicates a problem communicating with the netserver or ILS server.

Possible Causes

- Web Services is unable to communicate with the ILS.
- The ILS server was temporarily unavailable (because of a restart to initialize configuration changes, for example).
- An idle connection was dropped by an intervening firewall.

Solutions

- Make sure the netserver and ILS are running.
- Check network connections.
- Set the Keep Alive Timer to enable keep alive requests. For more information, see [Fields: ILS Configuration on page 48](#).

HatIOException: Read of response timed out

This error indicates that a request to the ILS did not get a response before the socket read timed out.

Possible Causes

- The request took longer to complete on the ILS than the socket read timeout allows.

Solutions

- Increase the socket read timeout. For more information, see [Fields: ILS Configuration on page 48](#).

HatSessionException: Unable to acquire session with server

This error indicates a problem establishing a session with the netserver.

Possible Causes

- Web Services is unable to establish a connection with the ILS, possibly because of a dropped connection.
- A socket read is timing out before a response is received.
- All sessions are busy.

Solutions

- Verify that the host name or IP address and port for the netserver are correct. For more information, see [Fields: ILS Configuration on page 48](#).
- Make sure the netserver and ILS are running.
- Set the Keep Alive Timer to mitigate dropped connections. For more information, see [Fields: ILS Configuration on page 48](#).
- Increase the socket read timeout. For more information, see [Fields: ILS Configuration on page 48](#).
- Increase the number of Maximum Sessions. For more information, see [Fields: ILS Configuration on page 48](#).

Initializing session manager failed

This startup error indicates a problem establishing a session with the netserver.

Possible Causes

- Web Services is unable to establish a connection with the ILS.

Solutions

- Verify that the host name or IP address and port for the netserver are correct. For more information, see [Fields: ILS Configuration on page 48](#).
- Make sure the netserver and ILS are running.

Initial LDAP bind failed

This error indicates a problem connecting to an LDAP server. The error includes the URL Web Services used to connect to the server.

Possible Causes

- The server is inaccessible.
- Required configuration values, such as host or port, are invalid.
- The server does not allow anonymous searching, or the BIND distinguished name or BIND Password you specified are invalid.

Solutions

- Make sure the server is available and is not blocked by firewalls or other network issues.
- Verify that the host name or IP address and port for the LDAP server are correct. For more information, see [LDAP Setup on page 60](#).
- Verify that the values for Distinguished Name for BIND and BIND Password are valid. For more information, see [LDAP Setup on page 60](#).

listenerStart error

This startup error indicates a JAXB incompatibility. The service will fail to start.

Possible Causes

- This error is commonly caused by using an earlier version of the JAXB API. JAXB 2.1 is required.

Solutions

- Make sure you are using the correct Java version. For more information, see [Java software on page 4](#).

Appendix A: Key Concepts

This section explains various Web Services concepts that are important to understand when configuring and managing Web Services.

See the following topics for more information:

Web Services base URL

The base URL is made up of the host name and port for your Tomcat and the Web Services name that you specified when you installed Web Services.

For example, if your Tomcat host name is *libraryapps.example.org* and you used the default port (8080) and the default Web Services name (*symws*), the base URL for your Web Services instance would be:

```
http://libraryapps.example.org:8080/symws
```

Appendix B: Advanced Tomcat configuration

The default Tomcat settings delivered with the installer are sufficient for most installations. The following JVM options are recommended when running Web Services as a single Tomcat instance:

```
-server  
-Xms128m -Xmx256m  
-XX:+UseParallelGC  
-Xss128k -XX:PermSize=128m
```

If you used the Tomcat distribution included in the Web Services installer, the `catalina.bat` (Windows) or `catalina.sh` (Linux or Solaris) file has already been modified with these options.

If you are running multiple instances of Web Services in Tomcat, you need to increase the minimum and maximum heap size (`-Xms` and `-Xmx`) by 50–75% (depending on anticipated traffic) per additional Web Services instance. If you are running on a 64-bit architecture and you find response times to be slow or you receive an out-of-memory error, you may need to double the stack size (`-Xss`).

However, depending on your system's configuration, your library's requirements, and other factors, you may want different options entirely. Tomcat has many configuration options, including advanced settings for heap space, stack space, perm space, threading options, and many security options. For details on configuring advanced Tomcat settings, please see the Tomcat documentation.

Appendix C: Managing password lockout

Staff and patron accounts are protected from hackers' brute force threats to access the accounts. A lockout system prevents hackers from easily cracking a password by locking the account from access after a specified number of failed attempts. Two levels of security can be implemented: a temporary lockout and a permanent lockout. These topics describe how to configure the account lockout feature and how the lockout can be cleared by patron and staff users.

For more information, see these topics:

Configuring password lockout	96
Clearing the lockout cache	97
Fields: Lockout Settings	97

Configuring password lockout

You can set up web services to lockout a user after a number of successive failed login attempts. There are two levels, temporary and permanent.

The temporary lockout occurs after a specified number of failed logins, and clears itself after a specified number of minutes where the user does not attempt to log in. After the time out expired, the user can try again.

The permanent lock is an absolute count. Each time the user fails a login, the count increments. When the maximum attempts is reached, login is disabled for that user until the user is reset, as explained in [Clearing the lockout cache on page 97](#).

To configure password lockout

1. Log in to the Admin console.
2. Click **Lockout Settings** in the Navigation pane.
3. Enter the values for each lockout property. For more information, see [Fields: Lockout Settings on page 97](#).

Note: Setting any of the fields to "0" turns password lockout off for that user type.

4. When you have finished, choose **Save**.

Related topics

[Appendix C: Managing password lockout on page 96](#)

[Clearing the lockout cache on page 97](#)

[Fields: Lockout Settings on page 97](#)

Clearing the lockout cache

Both the temporary and permanent lockout caches are cleared when the user either logs in successfully using the same ID, using a different ID (such as with a barcode instead of the preferred ID), or using the "forgot my PIN" feature. Otherwise, an administrator can clear the lockout user's web services caches. When any of these remedies are completed, the user's lockout counters are reset to 0.

The temporary and permanent locks can be reset in a number of ways:

- The user successfully logs in.
- The user resets the password through the "Forgot Password" link.
- A library worker resets the lockout cache for all users by clearing the general lockout cache (for more information, see [Clearing the lockout cache on page 97](#)).
- An administrator restarts Web Services or resets all of the Web Services caches. This resets the lockout cache for all users.

Each of these reset actions resets the count for both the temporary and permanent locks.

The permanent lockout cache can also be reset by a staff user through the patron resource or staff resource. See the *Web Services for Horizon and Symphony Software Developers Kit* for more information.

Related topics

[Configuring password lockout on page 96](#)

[Appendix C: Managing password lockout on page 96](#)

[Fields: Lockout Settings on page 97](#)

Fields: Lockout Settings

You can set up web services to lockout a user after a number of successive failed login attempts. There are two levels, temporary and permanent.

The temporary lockout occurs after a specified number of failed logins, and clears itself after a specified number of minutes where the user does not attempt to log in. After the time out expired, the user can try again.

The permanent lock is an absolute count. Each time the user fails a login, the count increments. When the configured maximum is reached, login is disabled for that user until the lock is removed for that user or the locks are removed for all users.



Setting any of the fields to "0" turns password lockout off for that user type.

Staff

Specifies the lockout settings for library worker accounts. The settings work the same as the Patron settings. Because of this, you can make the Staff lockout settings more or less restrictive, as needed.

Attempts before lockout

Specifies the number of times the wrong password can be entered for the account before the account is temporarily blocked (for the time specified).

Max attempts before total lockout

Specifies the number of times the wrong password can be entered for the account before the account is permanently blocked. A permanently blocked account must be reset by library system administrator.

Time locked out before retry

The amount of time, in seconds, minutes, hours or days, that must pass before the user can log in with a successful user ID and password.

Patron

Specifies the lockout settings for library worker accounts. The settings work the same as the Staff settings. Because of this, you can make the Patron lockout settings more or less restrictive, as needed.

Attempts before lockout

Specifies the number of times the wrong password can be entered for the account before the account is temporarily blocked (for the time specified).

Max attempts before total lockout

Specifies the number of times the wrong password can be entered for the account before the account is permanently blocked. A permanently blocked account must be reset by library system administrator.

Time locked out before retry

The amount of time, in seconds, minutes, hours or days, that must pass before the user can log in with a successful user ID and password.

Buttons

Remove all Locks

Resets all of the counters of users' attempted logins to zero (0). This does not reset or affect the values in the Lockout Settings. Login locks can also be reset on an individual basis through your ILS.

Related topics

[Configuring password lockout on page 96](#)

[Clearing the lockout cache on page 97](#)

Appendix D: Configuring email Templates

This section includes these topics:

Configuring Checkout Receipt Email Template	100
Configuring the Reset My PIN Template File	103

Configuring Checkout Receipt Email Template

Web Services for Symphony uses two files that together create the template the ILS uses to send checkout receipts via email. This topic contains information on how to customize these template files for use by your library. Both email template files are HTML-based. You can add to or edit the templates as you would any HTML file.



Your library system must have an email subsystem, such as blat or sendmail, set up on the ILS server to use this functionality. If you have a subsystem configured for emailing notice reports, the ILS will use this subsystem for emailing checkout receipts. For more information on setting up an email subsystem, contact SirsiDynix Customer Support.

The two default template files, `default.body` and `default.item`, are delivered with Symphony in the `Unicorn/Templates/EmailCheckoutReceipt/Default` directory. Along with the `Default` directory, Symphony delivers an additional directory for each delivered language. Symphony will use the template files in the directory appropriate to the specified language in the patron's Demographics details.



The template files are only delivered in the `Unicorn/Templates/EmailCheckoutReceipt/Default` directory. If you want to set up a default checkout receipt template for any delivered languages, you need to copy the template files in the `Default` directory to the directory for the delivered language and modify them according to your library's needs.



Also included in this directory are two additional files: `default.body.deliv` and `default.item.deliv`. These files are delivered with Symphony and should not be modified.

If Web Services for Symphony cannot find the template files with the specified name, it will check the language directory appropriate to the patron's language for the default template files. If there are no template files in the appropriate language directory, Web Services for Symphony will check the Unicorn/Templates/EmailCheckoutReceipt/Default directory for files with the name specified in the **Template** field; if it cannot find the files, it will use the default template files delivered with Web Services for Symphony.

The .body file

The .body file contains the template for email checkout receipts. The email message the ILS sends to patrons is based on the setup of the .body file.

Several of the first lines of the .body file contain comments on how to use the file. These lines do not show in the emails sent to patrons. The second line of the .body file, however, contains the return address for the checkout email. The default address is noreply@sirsidynix.com.

Using tags

The following tags are included in the default.body file to deliver the appropriate information to your patrons. The Web Services for Symphony server fills these placeholders with the appropriate values. You can place these tags wherever you want.

Tag	Description
<NAME>	The name of the user.
<USERID>	The user ID of the user.
<ITEMS>	The information for the checked out items. Important: The <ITEMS> tag is responsible for supplying the checkout receipt with the list of checkouts; in order for checked out items to display in checkout receipts, the <ITEMS> tag must be in the template.

Using labels

The .body file can also use Symphony labels to populate tags. The following labels are used in the delivered default.body file, but you can change which labels the file uses; alternatively, you can remove these labels and replace them with plain text, if desired.

Tag	Description
<code><charge_receipt_subject></code>	The subject line of the email.
<code><charge_receipt_header></code>	The header of the email.
<code><receipt_field_title:c></code>	The heading of the title column.
<code><receipt_field_author:c></code>	The heading of the author column.
<code><receipt_field_call_number:c></code>	The heading of the call number column.
<code><receipt_field_item_id:c></code>	The heading of the item ID column.
<code><receipt_field_date_charged:c></code>	The heading of the date charged column.
<code><receipt_field_date_due:c></code>	The heading of the date due column.
<code><charge_receipt_footer></code>	The footer of the email.

You can also use other labels from the `labels` file on the Symphony server to customize the `.body` file.

Be sure to save your changes once you are finished editing the file; any changes will be reflected immediately in subsequent email checkout receipts.

The `.item` file

The `.item` file contains the template for each row in the list of checked out items; the list this file creates displays where the `<ITEMS>` tag in the `default.body` file is located.

The following tags are delivered in the `default.item` file; the Web Services for Symphony server fills these placeholders with the appropriate values.

Tag	Description
<TITLE>	The title of the checked out item.
<AUTHOR>	The author of the checked out item.
<CALLNUM>	The call number of the checked out item.
<ITEMID>	The item ID of the checked out item.
<CHECKOUTDATE>	The date the checked out item was charged.
<DUEDATE>	The date the checked out item is due back to the library.

Configuring the Reset My PIN Template File

This topic contains information on how to customize the reset PIN template file for use by your library. The email template file is HTML-based. You can add to or edit the template as you would any HTML file. You can find the template on the Web Services for Symphony ILS server: `Unicorn/Templates/PINRecovery/Default`.

If Web Services for Symphony cannot find the template files with the specified name, it will check the language directory appropriate to the patron's language for the default template files. If there are no template files in the appropriate language directory, Web Services for Symphony will check the `Unicorn/Templates/PINRecovery/Default` directory for files with the name specified in the **Template** field; if it cannot find the files, it will use the default template files delivered with Web Services for Symphony.

You can set up the PIN recovery email by using any of the following tags:

- `<LINK>` — tag will be replaced with the newly formatted link. This is the `resetPinURL` given in this request, plus, if `<RESET_PIN_TOKEN>` was included in the URL, the system converts the `<RESET_PIN_TOKEN>` to a `resetPinToken`. Typically, this is used instead of `<PIN>`.
- `<RESET_PIN_TOKEN>` — tag will be replaced with the newly generated `resetPinToken`. Typically, this is included in the `resetPinURL` as part of this request instead of in the email template file.
- `<PIN>` — tag will be replaced with the user's current PIN. Typically, this is used instead of `<LINK>` to show the user their current PIN. This is useful for sites that do not want users to be able to change PINs. However, since this value gets sent in an email to the user, there may be a security risk.

The first line of the email template file should contain the reply-to address. The second line should contain the subject line of the email. The remainder of the template file contains the email body, including any of the above tags.



Symphony uses the `<CODE>` tag in the email template instead of `<RESET_PIN_TOKEN>`. Horizon Web Services and Web Services for Horizon and Symphony use `<RESET_PIN_TOKEN>`, but Web Services for Symphony also recognizes both `<CODE>` and `<RESET_PIN_TOKEN>`.

The following is an example of a template:

```
replyto@yourlibrary.com
PIN RESET
<html>
  <body>
    Please follow this link to <a href="<LINK>">reset your PIN</a>.<br/>
    From,<br/>
    <br/>
    Your local library
  </body>
</html>
```



As with the patron login credentials, `/user/patron/resetMyPin` includes separate fields that can be for authenticating. These include login, preferredID, alternateID, and barcode.

Index

/

- /circulation/block resource 70
- /circulation/block/detailLine resource 71
- /user/patron resource 72
- /user/patron/address resource 76
- /user/patron/address1 resource 76

A

- About the Resource Security Configuration 69
- Admin console
 - accessing 40, 83
 - Add/Edit LDAP Server (fields) 64
 - Add/Edit Single Sign-on URL (fields) 58
 - browser requirements 40
 - changing the password 44
 - changing the username 43
 - Client IDs page (fields) 53
 - configuring CAS single sign-on 57
 - help 42
 - ILS Configuration page (fields) 48
 - LDAP Setup 60
 - LDAP Setup page (fields) 63
 - logging in 42
 - logging out 43
 - managing Client IDs 52
 - message bar 42
 - Navigation pane 41
 - Patron Primary Address Configuration 67
 - Patron Primary Address Configuration page (fields) 67
 - resetting ILS connections 45
 - Resource Security Configuration 68
 - restoring access 83

- Single Sign-on Setup page (fields) 57
- Status page 44
- Status page (fields) 46
- understanding the basics 40
- understanding the interface 41
- updating ILS configuration options 48
- viewing Web Services logs 77
- workspace 42
- Admin console buttons
 - Add LDAP Server 64
 - Add URL (CAS) 58
- admin password
 - changing 44
 - lost 83
- admin username
 - changing 43
 - lost 83
- advanced installation
 - creating the properties file 27
 - introduction 27
 - running the installer 37
- authentication methods
 - CAS 57
 - LDAP 60

B

- base URL 94
- BeanCreationException error 89
- bind exception error 89
- browser requirements 40

C

- CAS
 - configuring single sign-on 57
- catalina log file 86
- configuring email checkout receipt tem-

plate files 100
context initialization error 90

E

errors

BeanCreationException 89
BindException 89
connection refused 90
context initialization failed 90
HATIOException 90
Initial LDAP bind 92
initializing session manager failed 92
listenerStart 93
port already in use 89
read of response timed out 91
unable to acquire session 91

F

fields

Always Require Authentication 51
Base Distinguished Name 63, 65
BIND Password 66
CAS Property 58
CAS Server 58
CAS Service ID 58-59
Client ID 53
Confirm BIND Password 66
Currency Code (Circulation) 51
Distinguished Name for BIND 65
ILS Connection Status 46
ILS Host 48
ILS Timezone 51
ILS Version 47
Keep Alive Timer (seconds) 50
LDAP Host 63-64
LDAP Only Authentication 64
LDAP Port 65
Licensed Clients 46
Licensed To 47
Maximum Idle Time (seconds) 50
Maximum Sessions 50
Minimum Sessions 50

netserver Port 48
Search Attribute 65
Socket Read Timeout
(milliseconds) 50
SSL Authentication (LDAP) 65
Station Login Clearance 49
Station Login Environment 49
Station Terminal Name 49
Station User Access Code 49
Station User ID 49
Symphony Interface 53
URL (CAS) 59
Web Services Version 46

Full Installation

steps 11

H

HATIOException error 90

help guide

about vii

I

ILS

managing Client IDs 52
resetting connections 45
supported versions 1
updating configuration options 48

ILS connection

troubleshooting 84

ILS properties

host 21, 48
netserver port number 21, 48
station access 23, 49
station clearance 22, 49
station environment 23, 49
station terminal 22, 49
station user ID 22, 49

Initial LDAP bind exception 92

initializing session manager error 92

installation

advanced installation 27
before you begin 7

- changing the locale 9
- configuration properties 19
- Console mode 8
- Full Installation 11
- getting started 1
- GUI mode 8
- ILS configuration properties 21
- installation types 10
- installing Web Services only 13
- installing Web Services with Tomcat 11
- overview 6
- running the installer 8
- Silent mode 27, 37
- starting up Tomcat 83
- starting your installation 7
- supported locales 10
- Tomcat configuration properties 19
- troubleshooting 26
- upgrading Web Services 15
- Web Services configuration properties 23
- installation type
 - choosing 10
 - Full Installation 11
 - Web Services Only 13
 - Web Services Upgrade 15
- installer
 - configuration properties 19
 - supported locales 10

J

- Java
 - environment variables
 - finding,setting 18
 - requirements 4
- JAVA_HOME
 - finding,setting 18
- JRE_HOME
 - finding, setting 18
- JVM_BIND exception 89

K

- key concepts
 - base URL 94

L

- LDAP 60
 - about LDAP authentication 60
 - changing timeout settings 62
 - managing server connections 61
- LDAP bind exception 92
- license keys
 - viewing or updating 45
- listenerStart error 93
- locales 9
- logging
 - catalina (Tomcat) 86
 - symws activity 87
 - Web Service requests 86
- logging in 42
- logging out 43
- logs
 - examining log files 86
 - viewing Web Services logs 77

O

- online help
 - opening 42
- operating system
 - requirements 4

P

- Patron Primary Address Configuration 67
- properties file
 - admin-settings properties 36
 - creating 27
 - essential properties 28
 - hat-session properties 32

R

- read timed out error 91

requests log file 86
Resource Security Configuration 68
 /circulation/block resource 67, 70
 /circulation/block/detailLine
 resource 71
 /user/patron resource 72
 /user/patron/address1 resource 76
 /user/patron/address2 resource 76
 /user/patron/address3 resource 76
About the Resource Security Con-
figuration 69
Setting resource security 69

S

sessions
 timeout 43
Setting resource security 69
single sign-on 57
 configuring 57
SSO 57
Symphony ILS
 supported versions 1
symws log file 87
system requirements 2

T

timeouts
 session 43
Tomcat
 advanced configuration 95
 catalina log files 86
 configuration 3
 configuration properties 19
 memory requirements 3
 ports 89
 recommended start up settings 95
 requirements 4
 session timeout 43
 starting up 83
Tomcat properties
 HTTP port 20
 HTTPS port 21

 service name 20
 shutdown port 21
troubleshooting 82
 Admin console access 83
 common errors 89
 connection issues 84
 examining log files 86
 general 82
 installation 26

U

unable to acquire session error 91
uninstallation 81

W

Web Services
 requests log file 86
 symws log file 87
Web Services for Symphony
 about 1
 Admin console 39
 configuring 39
 hardware requirements 3
 installing 6
 Java requirements 4
 key concepts 94
 license keys 45
 run-time status 44
 supporting operating systems 4
 system requirements 2
 Tomcat requirements 4
 troubleshooting 82
 uninstalling 81
 upgrading 15
 verifying that Web Services are
 running 85
Web Services Only Installation
 steps 13
Web Services properties
 currency code 25, 51
 default language code 25
 keep alive timer 25, 50

- logging directory 26
- logging prefix 26
- maximum session idle time 24, 50
- maximum sessions 24, 50
- minimum sessions 24, 50
- session time out 25, 50
- time zone 51
- web services requirements
 - hardware 3
 - Java requirements 4
 - operating system 4
 - system 2
 - Tomcat requirements 4
- Web Services Upgrade
 - steps 15