OCTOBER 2019

Release Notes

Web Services for Horizon and Symphony 6.0.0

SirsiDynix®

Contents

Overview

These Web Services for Horizon and Symphony 6.0.0 Release Notes cover enhancements and bug fixes since Web Services for Horizon and Symphony 2019.03.2, as well as system requirements and frequently asked questions for version 6.0.0.

> ⚠ This version of Web Services for Horizon and Symphony deprecates the version indicator ("v1") in request URIs. Enterprise and Portfolio versions 4.5.1 and 5.0 contain the "v1" endpoints. SirsiDynix strongly recommends that until you can update your library to Enterprise 5.0.1, do not update your Web Services version beyond 2019.03.2.
>
> The current version of MobileCirc includes some "v1" endpoints that don't work with Web Services for Horizon and Symphony 6.0.0. A refactored version of MobileCirc has been tested and works well with 6.0.0. If your site uses MobileCirc, SirsiDynix recommends that you wait to update Web Services until after the planned December 2019 release of MobileCirc with BLUEcloud Central.

Refer to individual sections in the release notes for more detailed information about other enhancements and fixed bugs.

> ℹ For answers to common questions about the features and requirements of Web Services for Horizon and Symphony, please see the FAQ section of this guide. For more information about installing Web Services for Horizon or Web Services for Symphony, or about upgrading your web services instance, see either the *Web Services for Horizon 6.0.0 Setup and Configuration Guide* or the *Web Services for Symphony 6.0.0 Setup and Configuration Guide*, depending on which version of web services your library currently uses. For any additional questions, please contact SirsiDynix Customer Support.

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.

Enhancements

This section contains information about key features and enhancements in Web Services for Horizon and Symphony 6.0.0. Enhancements are modifications to the software from the previous version. The following list includes version 6.0.0 enhancements, with a brief description of how each enhancement improves functionality.

This chapter includes these sections:

Web Services for Horizon and Symphony

Removal of patron editable fields

In specific fields in the Resource Security Configuration in Web Services for Horizon and Symphony Admin console, the option for patrons to edit some fields has been removed.

Ability to turn off legacy web services

Administrators can now disable legacy web services calls in order to determine if their client is still using them.

Searching bibs by index

Web Services for Horizon and Symphony now lets users search for bibs by index.

Default login lockout settings tightened

In order to help libraries transition to account lockout restrictions, the default settings have been set to 20 failed attempts before a lockout happens, and 5 minute lockout is imposed when the maximum attempts have been met. SirsiDynix recommends changing these settings to be more secure at the earliest opportunity.

Staff authentication logs

A new log file (`hzws-securityaudit.log`, `symws-securityaudit.log`) has been added to track staff login attempts through Web Services. The log gathers the following data:

- Timestamp
- ID used

- Originating IP
- HTTP response code
- Client ID
- Originating app ID

Logging when a staff user accesses patron information

Web Services now logs when a staff user accesses the information in a patron's account. This includes the following information:

- Timestamp
- Staff user's ID
- Client ID
- Originating app ID
- The full text of the request (not including the request body or resulting data)
- IP address of originating request
- HTTP response code
- Relevant header info

Web Services for Horizon

Using the significant digits of a patron barcode for login

Web Services for Horizon now supports patron logins using only the significant barcode digits. To activate this feature, an administrator must set up the Borrower Barcode Prefix and Borrower Barcode Length in the Location table and enable Allow Patron Login without Barcode Prefix in the ILS Configuration in Web Services for Horizon.

Handling non-monetary payments

The addDetailLine action no longer logs payments of "0" when the payment has no monetary value.

Handling control characters in title and author fields

Web Services for Horizon now removes any control characters from an item's Title and Author fields before returning the bibliographic information in a response.

Filling holds based on item status

Clients can now use any item to fill a hold if the item's status is currently a status in which available_for_hold is set to "true". However, if the status would require Web Services for Horizon to do additional processing to change the status (such as "lost" or "claimed returned") it will return an error.

Improved performance when obtaining blocks

The `/user/patronStatusInfo` resource has been refactored to be more efficient when obtaining block information.

Simplification of detail line request

The patron field is no longer required for the `/circulation/block/detailLine` request.

Support for end-of-day loan periods

Clients will be able to take advantage of Horizon's ability to define a special Circulation rule for a Location group that makes items of a specific BType, IType (or both) due when the library closes.

New action to display the hold shelf list

The /circulation/holdRecord resource now includes the `displayHoldShelfList` action which returns the items that are currently on the hold shelf. Staff users can use the displayStatus enum to view all items, only the active items, or only the expired items.

Requesting a patron block history

The `/user/patron` resource can now include the list of blocks that have been made against the patron's account.

Additional information in library policy requests

Web Services for Horizon now returns the following information with the `/policy/library` request:

- Daily hours
- Holidays hours
- Library address
- Special due dates
- Others

Accessing policy data for bib and auth validation

The following Horizon tables can now be accessed through web services:

- cat_element
- cat_element_value
- cat_help
- cat_indicator
- cat_indicator_value
- cat_subfield
- cat_tag
- cat_tag_group
- cat_tag_link

Serials policies

The following policies have been made available to clients to enable serials management:

- bindingClaimType
- bindingFoilColor
- bindingFont
- bindingMaterial
- bindingMaterialColor
- bindingMethod
- bindingStatus
- bindingTrigger
- chronPattern
- publicationPattern
- runCode

Driver license field in patron resource

The `/user/patron` resource now includes a field for the borrower's driver license number.

Improved support for offline files

The Admin console in Web Services for Horizon now lets libraries view the asset files that are created by offline transactions. Administrators can see the file names and properties of offline asset files and delete those that are no longer needed.

Improved Admin console interface

To streamline the Web Services for Horizon Admin console interface, certain extraneous fields have been removed. This does not affect Web Services functionality.

Item circulation history tracking

A new resource, `/circulation/itemCircHistory`, has been added that records the circulation details of an item after it has been checked in.

Improved password security

A number of resources were added so clients can take advantage of the new security upgrades that will be available in Horizon 7.5.6.

Increased transaction accuracy across timezones

Web Services for Horizon now bases all transactions on the time zone set in the Web Services Admin console rather than on the timezone of the server. This ensures that transactions are more accurate.

Web Services for Symphony

Creating items from MARC holdings

Clients using Web Services for Symphony can now create item records associated with MARC holding records.

Searching patrons by library or library group

The `/patron/search` now includes the libraryGroup field to let clients search for patrons by library or library group.

Data validation when creating a bib with MARC tags

Web Services for Symphony now supports data validation for MARC tags when creating or editing a bib record.

Support for academic reserves and offline transactions

Web Services for Symphony now supports offline checkouts of academic reserves items.

Better access to localized strings

Clients can now request the translated string of a policy label using either the policy label string or the message number.

Support for bill history

Clients can now retrieve a list of a library patron's bills with bill and payment details.

Requesting patron inactive holds

When requesting the `user/patron/holdRecordList`, clients can now request a patron's inactive holds.

Improved performance when requesting delinq user information

Web Services for Symphony now caches the HAT response the first time a request for the delinquent user information is made and fills subsequent requests from the cached value. This reduces the number of calls needed to process delinquent user requests and improves processing times.

Labels for policy description translations

All of the policy resources now include a `translatedDescription` field for translations of the resource description.

Additional self-registration fields

Web Services for Symphony now includes these additional fields for verifying that a new self-registration is not a duplicate:

- Birth date
- Email address
- Phone number

Ability to report refund information

The `finesAndFeesRefunded` has been added to the `/circulation/circRecord/checkIn` action to return the amount of a refund when a patron returns an item whose status was LOST.

Item information added to block resource

The fields for the title, call number, and item library have been added to the `/circulation/block` resource to preserve billing information after a bib has been deleted.

Consortium policies accessible through web services

The Symphony consortium policies, Consortium Group CGRP, Consortium Membership CMBR, and Library Display Profile DPRO, are now accessible through Web Services for Symphony.

Hold pull requests more efficient

Web Services for Symphony now takes advantage of the efficiencies in Symphony 3.6.1 when requesting a hold pull list.

Accessing the reserve list

The `circulation/circRecord/checkIn` and `circulation/itemCircInfo/advise` actions now include the `reserveList` field to return the list of reserves.

Additional fields in the hold record resource

The `/circulation/holdRecord` resource now includes fields for the notice date, whether the hold was placed using an override, and whether the hold was made for a reserve item.

Ability to fill a reserve hold

Web Services for Symphony can now trap reserve items for holds when they are checked in.

Improved matching of entry numbers

The handling of MARC updates has been improved to better match records through the Format Policy. This ensures that internal entry numbers are correctly matched to entry IDs.

Rental fee prompt

If an item includes a fee for checking it out, Web Services clients now prompt for the fee to be paid.

Renewal fee prompt

When renewing the checkout of an item, Web Services for Symphony now checks with the Symphony ILS to determine if a fee should be charged. When positive, Web Services sends a prompt to bill for the fee. When negative, no prompt is sent.

Ability to change the on-shelf hold expiration date

Web Services for Symphony clients can now use a request to change the on-shelf expiration date of an active, expired hold, which returns the status to being held.

Improved alternateID validation when creating a user

When a staff user creates a patron account, Web Services for Symphony now validates that the alternate ID is not already in use. If it is, Web Services now returns an error message that better explains that the alternate ID is already in user rather than returning a Hat error message (hatErrorResponse.658).

Managing MARC holdings

Clients using Web Services for Symphony can now create, edit, and delete MARC holdings on the SirsiDynix Symphony ILS.

Serials policies

The following policies have been made available to clients to enable serials management:

- ACRN
- CLAIM_TYPE
- FREQ
- HOLI
- PPCM
- PPLM
- SCLR
- SCRN
- SCST
- SCT1
- SCT2
- SERIAL_LABEL
- SERIAL_PATTERN
- SERXINFO_ENTRY_TYPE
- SRNM
- SRNT
- SRNT_SPECPROC
- VGR1
- VGR2

- VGR3
- VRES

Improved client ID management

The interface for the Client ID has been improved by removing functions that have been confusing. This includes the ability to add and delete client IDs, which did not license the IDs. All of the clients licensed for the library's use are listed with the known client ID. If neither Web Services nor Symphony can identify the client, the library can select the client ID from a list. However, SirsiDynix recommends that you specify the client ID for the unknown client with the Symphony WorkFlows client (for more information, see the SirsiDynix Symphony WorkFlows Help).

Improved logging for lookupHoldPickupLibraryList action

The SymCirculationHelper logger now logs information that makes it easier to determine why a library might have been left off a pickup location list.

Increased display hold shelf response efficiency

The Display Hold Shelf List responses have been made more efficient by reducing the amount of data passed between the ILS and Web Services and requesting only the information that is needed for the current request.

Improved password management

The `/user/patron` resource now includes actions to allow patrons to change and reset their passwords.

Frequently asked questions

What products are compatible with Web Services for Horizon and Symphony 6.0.0?

Information about all compatible products is listed in the Compatibility information section of these release notes (see **Compatibility information** on page 18), regardless of whether a product uses the legacy versions of web services that are included in Web Services for Horizon and Symphony (for example, Enterprise) or if a product uses the new framework (for example, BLUEcloud Cataloging).

Can upgrading my instance of web services break compatibility with any current products?

Depending on the version of the product your library is currently using, upgrading to Web Services for Horizon or Web Services for Symphony could break compatibility with that product. Please review the Compatibility information section in this guide (see **Compatibility information** on page 18) before deciding whether to install or upgrade to Web Services for Horizon or Web Services for Symphony.

Do Web Services for Horizon need to be installed on the same server as HIP? (SirsiDynix Horizon libraries)

No, Web Services for Horizon do not need to be installed on the same server that runs HIP. However, if you will be using the legacy Horizon Web Services framework within Web Services for Horizon, then your Web Services for Horizon instance will need to communicate with HIP. If this is the case, you can provide connection details for your HIP server during the installation of Web Services for Horizon.

How can I keep my current web services settings when I install or upgrade?

Simply copy your settings files from your current web services directory to the directory with the installer executable file. The installer will use any web services settings files that are included in its directory to pre-populate those settings during installation.

> ⚠ Depending on your current version of web services, you might need to uninstall both the legacy version of web services and Tomcat and then perform a full installation with the new installer.

For information about installing Web Services for Horizon or Web Services for Symphony, or about upgrading your web services instance, see either the *Web Services for Horizon 6.0.0 Setup and Configuration Guide* or the *Web Services for Symphony 6.0.0 Setup and Configuration Guide*, depending on which version of web services your library currently uses.

What languages are supported with this release of Web Services for Horizon and Symphony?

The Web Services for Horizon and Symphony Admin consoles, installers, and exception messages for web services requests have already been localized into several languages (see the list below). We will continue to add localizations in future updates.

> ⚠ Do you want to see Web Services for Horizon and Symphony in another language? Do you have feedback about a translation in your native language? You can help! Contact us at docs@sirsidynix.com to offer feedback.

Web Services for Horizon and Symphony currently supports the following languages:

- Chinese (Traditional)
- Chinese (Simplified)
- English (United Kingdoms)
- English (United States)
- French (France)
- French (Canadian)
- Spanish

What is ROA and why should I be excited about it?

ROA (Resource Oriented Architecture) message structures are an alternative to the commonly used SOA (Service Oriented Architecture) message structures. ROA message structures provide a number of important features, including:

- Flexible web services technology

  ROA is used by many popular cloud & web based products such as Salesforce, JIRA, and Git. These companies and more choose ROA message structures for their increased flexibility, performance, and agility when accessing a highly dynamic set of data. Because of the inherent differences between SirsiDynix Horizon and SirsiDynix Symphony systems, ROA message structures are the perfect choice for Web Services for Horizon and Symphony.

- Improved, RESTful message structures

  ROA prescribes a more dynamic, RESTful message structure along with access to resource describing metadata and APIs.

- Generic containers

  ROA uses generic container structures for transferring object definitions and their data, allowing for more optimal component reuse. This optimization helps developers implement performance gains.

- Self describing resources and APIs

  Self-describing resources and APIs enable SirsiDynix to provide standardized metadata for all elements within a resource and its subordinates. This feature allows different clients to be more consistent in their interaction with resources.

- Flexible object definitions

  For SOA, object definitions tend to be strict contracts that break as values are added or removed. ROA uses dynamic structures that do not break contract as objects change. This flexibility allows clients to inspect and adapt to resource variations due to back-end version differences, configuration settings, or even customizations.

- Targeted value retrieval

  ROA gives clients the ability to retrieve only the values they need, reducing processing time in the code and reducing bandwidth over the network. For example, some processes that formerly required up to 50 separate requests can now be accomplished in a single call.

System requirements

Web Services for Horizon and 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 HIP server (SirsiDynix Horizon), ILS server (SirsiDynix Symphony), or another server (either ILS).

ILS version requirements

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

- SirsiDynix Symphony 3.5.3, 3.6, 3.6.1, 3.6.2

  Note: SirsiDynix Symphony 3.6.2 is required for Web Services for Symphony to be fully functional; some functionality may not work until Symphony is upgraded to this version.

- SirsiDynix Horizon 7.5.4, 7.5.4.1, or 7.5.5

  Note: SirsiDynix Horizon 7.5.5 is required for Web Services for Horizon to be fully functional; some functionality may not work until Horizon is upgraded to this version.

> For information about installing Web Services for Horizon or Web Services for Symphony, or about upgrading your web services instance, see either the *Web Services for Horizon 6.0.0 Setup and Configuration Guide* or the *Web Services for Symphony 6.0.0 Setup and Configuration Guide*, depending on which version of web services your library currently uses.

Tomcat requirements

To use Web Services for Horizon and Symphony, you must also have Apache Tomcat version 8.5 or higher (a distribution of Tomcat version 8.5.46 is included in the installer for your convenience).

> 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.

SirsiDynix recommends using the Tomcat version included with the installer for the best performance and compatibility with Web Services for Horizon and Symphony.

> Remember to uninstall any previous versions of Tomcat before installing the latest version that is included with the installer.

Hardware

Web Services for Horizon and Symphony supports 64-bit architectures.

During typical operation, Web Services for Horizon and 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 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 either the *Web Services for Horizon 6.0.0 Setup and Configuration Guide* or the *Web Services for Symphony 6.0.0 Setup and Configuration Guide*, depending on which version of web services your library currently uses.

Operating system

Web Services for Horizon and 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)

Java software

Java 8

Web Services for Horizon and Symphony 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

Compatibility information

The following list details which product versions are compatible with Web Services for Horizon and Symphony 6.0.0. For questions about ILS requirements, see the System requirements section in this guide. If you have additional questions about the compatibility of another product, please contact SirsiDynix Customer Support.

Compatible Products

- BLUEcloud Cataloging
- BLUEcloud Circulation
- BLUEcloud Mobile
- BLUEcloud PAC
- BookMyne
- Enterprise 5.0.1
- MobileCirc for Horizon (following the December 2019 BLUEcloud Central release)
- MobileCirc for Symphony

Deprecation Notices

This section gives notice of portions of the Web Services for Horizon and Symphony for which support will be discontinued in future releases. It lists the portion of the web services that will be deprecated, including the time frame when support will be dropped.

Version indicator ("v1") in the request URI

In previous versions of Web Services for Horizon and Symphony, Web Services required that "v1" or "ws" be included in the URI request. In this 6.0.0 release, the "v1" has been deprecated. You should not upgrade to this version unless you have completely removed all "v1" endpoints.

SirsiDynix Enterprise and SirsiDynix Portfolio support

Enterprise and Portfolio versions 4.5.1 and 5.0 contain the "v1" endpoints. SirsiDynix strongly recommends that until you can update your library to Enterprise 5.0.1, do not update your Web Services version beyond 2019.03.2.

MobileCirc support

The current version of MobileCirc includes some "v1" endpoints that don't work with Web Services for Horizon and Symphony 6.0.0. A refactored version of MobileCirc has been tested and works well with 6.0.0. If your site uses MobileCirc, SirsiDynix recommends that you wait to update Web Services until after the planned December 2019 release of MobileCirc with BLUEcloud Central.

Legacy Horizon Web Services and Symphony Web Services

Currently, Web Services for Horizon and Symphony works with some legacy web service calls. However, this support will eventually be deprecated in a future version of Web Services for Horizon and Symphony. SirsiDynix recommends that you begin to locate instances where legacy Horizon Web Services and Symphony Web Services calls are referenced and replace them with the ROA version of those web services calls. To help with this migration to ROA web services, the Admin console now includes an option to disable legacy support. For more information, see "Fields: ILS Configuration" in the Admin console Help.

Fixed bugs

This section contains a list of all fixed bugs for Web Services for Horizon and Symphony 6.0.0. Fixed bugs are problems that SirsiDynix has identified through the testing process and fixed since Web Services for Horizon and Symphony 2019.03.2. The following table includes a list of fixed bugs included in version 6.0.0. Each fixed bug indicates whether it affects SirsiDynix Horizon or SirsiDynix Symphony and has a corresponding tracking number and a brief description of how it has been fixed.

ILSWS-2115 **Horizon: Patron subresources always including some fields**

Some fields in the patron resource were being included even when they were explicitly requested to be excluded. These included the number on the phone resource, the barcode on the barcode resource, and all fields in the `primaryPatronAddress` resource. This has been fixed to exclude these fields when not wanted in the response.

ILSWS-2566 **Horizon: Patron search wildcards**

Performing a patron wildcard search was returning invalid results. This has been fixed to return only results that match the wildcard search.

ILSWS-2649 **Symphony: Register without pin error**

If patron self-registration was configured to allow patrons to create an account without a PIN or password, when the client attempted to create a session token for a patron, an error would occur. This has been fixed to not automatically attempt to create the session token under these circumstances. Instead, patrons must use "Forgot Password" to create a new PIN or password.

ILSWS-2918 **RSC policy entry describes**

Requests that included `policyList` in the `resourceSecurityConfig` would not return any of the `blockEntry` fields. This has been fixed by defining two distinct resources, `/adminws/resourceSecurityConfig/blockResourceEntry` and `/adminws/resourceSecurityConfig/blockPolicyEntry`.

ILSWS-2960 **Horizon: Patron search describing numeric searches as string**

The `user/patron/search` describe listed all of the numeric fields as "string" rather than as numeric. This has been fixed to list the correct field types.

ILSWS-2974 **Horizon: The patronStatusInfo resource has maximum when count and max are 0**

In `/user/patronStatusInfo`, if a maximum (such as finesMax) and count (such as fines) were both zero, the associated hasMaximum (such as hasMaxFines) field would remain "true". This has been fixed to change the `hasMaximum` field to "false" when the count is zero.

ILSWS-3001 **Horizon: itype exceptions not handled for circ privilege 11 Max Items Out**

When exceptions to the default rule were included, the itype exceptions were not handled for circ privilege 11 Max Items Out. The checkout was allowed without an override using Web Services, however, in the Horizon desktop client, an override was required. This has been fixed to also require an override when using Web Services.

ILSWS-3003 **Horizon: Diacritics corrupted in long text fields**

When a diacritic character appeared in a large text field, it would become corrupted when displayed. This has been fixed to display diacritic characters correctly in these fields.

ILSWS-3043 **Symphony: Creating a patron using AUTO for barcode will generate two active IDs**

When using a POST method with the `/user/patron` resource to create a new patron record, if "AUTO" was used for the barcode field, the patron record would contain two active IDs. This has been fixed to create only one active ID when "Auto" is used in the barcode field.

ILSWS-3053 **DHC.json from SDK download no longer loads into restlet client**

The Talend API Tester (formerly DHC) no longer imports the DHC.json template file. SirsiDynix suggests that you use a different rest client, such as Postman, to test Web Services for Horizon and Symphony. The `DHC.json` file has been renamed (`postman-horizon.json` or `postman-horizon.json`) and moved to a new directory. You can find the file in the SDK at the Files and Libraries link. Extended documentation of the SDK can also be found in the SDK at the Web Services SDK link.

ILSWS-3121 **Backslash in title causes library to be null in circRecordList**

If an item in which the title contained a backslash ("\") was checked out to a user, the library would show "null" for that checkout in the user's `circRecordList`. This has been fixed to display the title for the item in the circRecordList.

ILSWS-3141 **Cannot find reserves using search if course ID includes spaces**

If a reserves course had a space in its course ID, user would not be able to find the course in a search. This has been fixed to allow the course to be found when a space exists in the course ID.

Known issues

The SirsiDynix Customer Support Center lists known issues for this and previous versions of Web Services for Horizon and Symphony. Log in to support.sirsidynix.com, select **Documents** and click **Known Issues & Enhancements**. We add and update known issues in this location as we become aware of them or learn more about them.

Click the link below to open the SirsiDynix Support Center and browse open issues specific to the following product:

- Web Services for Horizon and Symphony