



# InterSystems IRIS Release Notes

Version 2019.4  
2020-01-28

*InterSystems IRIS Release Notes*

InterSystems IRIS Data Platform Version 2019.4 2020-01-28

Copyright © 2020 InterSystems Corporation

All rights reserved.

InterSystems, InterSystems IRIS, InterSystems Caché, InterSystems Ensemble, and InterSystems HealthShare are registered trademarks of InterSystems Corporation.

All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

This document contains trade secret and confidential information which is the property of InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142, or its affiliates, and is furnished for the sole purpose of the operation and maintenance of the products of InterSystems Corporation. No part of this publication is to be used for any other purpose, and this publication is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the express prior written consent of InterSystems Corporation.

The copying, use and disposition of this document and the software programs described herein is prohibited except to the limited extent set forth in the standard software license agreement(s) of InterSystems Corporation covering such programs and related documentation. InterSystems Corporation makes no representations and warranties concerning such software programs other than those set forth in such standard software license agreement(s). In addition, the liability of InterSystems Corporation for any losses or damages relating to or arising out of the use of such software programs is limited in the manner set forth in such standard software license agreement(s).

THE FOREGOING IS A GENERAL SUMMARY OF THE RESTRICTIONS AND LIMITATIONS IMPOSED BY INTERSYSTEMS CORPORATION ON THE USE OF, AND LIABILITY ARISING FROM, ITS COMPUTER SOFTWARE. FOR COMPLETE INFORMATION REFERENCE SHOULD BE MADE TO THE STANDARD SOFTWARE LICENSE AGREEMENT(S) OF INTERSYSTEMS CORPORATION, COPIES OF WHICH WILL BE MADE AVAILABLE UPON REQUEST.

InterSystems Corporation disclaims responsibility for errors which may appear in this document, and it reserves the right, in its sole discretion and without notice, to make substitutions and modifications in the products and practices described in this document.

For Support questions about any InterSystems products, contact:

**InterSystems Worldwide Response Center (WRC)**

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: support@InterSystems.com

# Table of Contents

<b>About This Book .....</b>	<b>1</b>
General Licensing Notes .....	1
Application Use Of InterSystems Web Server .....	1
<b>1 New and Enhanced Features for InterSystems IRIS 2019.4 .....</b>	<b>3</b>
1.1 Continuous Delivery Releases of InterSystems IRIS .....	3
1.2 SQL Enhancements .....	4
1.3 Infrastructure and Cloud Deployment Improvements .....	4
1.4 Native API for Java and .NET Enhancements .....	4
1.5 New Automatic Configuration Customization .....	4
1.6 Other Enhancements and Efficiency Improvements .....	5
<b>2 General Upgrade Information .....</b>	<b>7</b>
2.1 Important Considerations .....	7
2.1.1 Compatibility .....	7
2.1.2 Preview Release .....	7
2.2 Upgrade Specifics .....	8
2.2.1 Upgrading Containers .....	8
2.2.2 Classes .....	8
2.2.3 Routines .....	9
2.2.4 Cached Queries .....	9
2.2.5 Web Services and SOAP .....	9
2.2.6 Frozen Plans for SQL Queries .....	9
<b>3 Upgrade Compatibility Checklist for InterSystems IRIS 2019.4 .....</b>	<b>11</b>
3.1 Administrators .....	11
3.1.1 Improved Container with Nonroot Default User Requires Installation Change .....	11
3.1.2 Changes for Dejournaling .....	12
3.2 Developers .....	12
3.2.1 Custom Code Cannot Assume Root Privileges Running in Docker Containers .....	12
3.2.2 Changes in SQL TRUNCATE TABLE Transactions .....	12
3.2.3 TSQL Parser Changes .....	12
3.2.4 Language Bindings Changes .....	13
3.2.5 Changes to TCP /BINDTO Behavior .....	13
3.2.6 Changes to %SYS.PhoneProvider .....	13
3.2.7 Changes in Display of Host Status in Interoperability Productions .....	13
3.2.8 Changes in X12 Acknowledgement Documents in Interoperability Productions .....	13
3.2.9 Changes in syncresponses Array in BPLs for Interoperability Productions .....	13
3.2.10 Changes to EnsLib.EDI.Segment Class for Interoperability Productions .....	14
3.2.11 Changes in HIPAA Schema for Interoperability Productions .....	14
3.2.12 Changes in Handling Unexpected HL7 Names in Interoperability Productions .....	14
<b>4 Known Issues and Notes .....</b>	<b>15</b>
4.1 Cloud Manager Support for SUSE Linux .....	15



# About This Book

This book describes the major features that have been added to InterSystems IRIS® data platform 2019.3, as well as information needed to update custom code from InterSystems IRIS® 2019.3.

It contains the following sections:

- [New and Enhanced Features for InterSystems IRIS 2019.4](#)
- [General Upgrade Information](#)
- [Compatibility Checklist for InterSystems IRIS 2019.4](#) — If you are accessing durable database or directories that have been used with a previous version of InterSystems IRIS® or are using custom code developed on an earlier version, read this section.
- [Known Issues and Notes](#) — Describes issues in InterSystems IRIS 2019.4 that you should be aware of and other notes related to the release.

## General Licensing Notes

InterSystems makes its products and features available under license to customers. While InterSystems may or may not enforce the use of said products or features consistent with the purchased license capabilities, customers are expected to comply with terms of their licenses. Moreover, InterSystems may tighten enforcement at any release without notice.

Developers must be aware that certain license types are required in order to use specific product features such as Multi-Server capability, Mirroring, and Web Services features. The specific requirements are noted in the InterSystems Price List and the Terms and Conditions for licensing. These are available from your local InterSystems representative.

## Application Use Of InterSystems Web Server

InterSystems installs an Apache-based web server (often referred to as the "private web server") to assure that the management portals for its products are always available. The private web server is built and configured to meet the management needs of InterSystems administrative servers and is configured to only connect to InterSystems products. The options selected are not, in general, suitable for production use - in particular, security is minimal and the options used are generally unsuitable for a high volume of HTTP requests. Testing, by InterSystems, of the private webserver only covers use of the private web server for managing InterSystems IRIS, HealthShare, and other InterSystems products.

Customers are not required to use this web server to manage our products. You may also use a web server of your choice, located on whatever server you elect to use. The private web server is provided as a convenience to simplify installation and installation dependencies. Many developers also find it useful to use the private web server for unit testing.

### UNIX®

The parameters used for the UNIX® build are:

```
--prefix=<installation_location>
--disable-actions
--disable-asis
--disable-auth
--disable-autoindex
--disable-cgi
```

```
--disable-cgid  
--disable-charset-lite  
--disable-dir  
--disable-env  
--disable-imap  
--disable-include  
--disable-negotiation  
--disable-setenvif  
--disable-shared  
--disable-status  
--disable-userdir  
--enable-access  
--enable-alias  
--enable-log-config  
--enable-mime  
--enable-so  
--without-berkeley-db  
--without-gdbm  
--without-ndbm
```

The server produced has defaults using the Apache Group's prefork Multi-Processing Module (MPM). This is the non-threaded server model. The number of requests that can be concurrently served is directly related to the number of Apache worker processes in the pool. The private web server is configured to occupy the smallest possible footprint by allowing a maximum of two worker processes to be created for the pool. The following settings will be found in the Apache configuration (`httpd.conf`) for the server:

- MinSpareServers: 1
- MaxSpareServers: 2

By contrast, the default Apache configuration for a production grade build is usually the following:

- StartServers: 5
- MinSpareServers: 2
- MaxSpareServers: 20
- ServerLimit: 256
- MaxClients: 256

This configuration will allow Apache to create 5 worker processes at start-up time, increasing to a maximum of 256 as the concurrent load increases. Because of these differences in configuration, the performance of the private web server will be noticeably inferior to that of a production grade Apache build as the concurrent load increases.

## Conclusion

If you expect very low volume of HTTP traffic, have limited demands for high availability and secure operation, the private web server may be suitable for your deployment needs. However, if you expect a high amount of HTTP traffic, require high availability in your web server, need to integrate with other sources of web information, or need a high degree of control over your web server, InterSystems recommends installing your own separate copy of Apache, ideally on its own server, and configuring it to use our Web Gateway to communicate with InterSystems products. Review the options above to determine if this is so.

# 1

# New and Enhanced Features for InterSystems IRIS 2019.4

This document describes the 2019.4 release of the InterSystems IRIS® data platform. The following sections describes this release and its new capabilities and enhancements:

- [Continuous Delivery releases of InterSystems IRIS](#)
- [SQL Enhancements](#)
- [Infrastructure and Cloud Deployment Improvements](#)
- [Native API for Java and .NET Enhancements](#)
- [New Automatic Configuration Customization](#)
- [Other Enhancements and Efficiency Improvements](#)

For a list of new features in the previous version, see the [InterSystems IRIS 2019.3 Release Notes and Upgrade Checklist](#).

## 1.1 Continuous Delivery Releases of InterSystems IRIS

InterSystems IRIS 2019.4 is a continuous delivery release of InterSystems IRIS in contrast with InterSystems IRIS 2019.1, which was an extended maintenance release. There are now two streams of InterSystems IRIS releases:

- Continuous delivery releases — These releases provide access to new features and are ideal for developing and deploying applications in the cloud or in local Docker containers.
- Extended maintenance releases — These releases are less frequent than the continuous delivery releases but provide the increased stability of maintenance releases. These releases are ideal for large enterprise applications where the ease of getting fixes in maintenance releases is more important than getting early access to new features.

Continuous delivery releases are provided in container format and are available on Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure, Docker Hub, and the InterSystems WRC download site. You can run a continuous delivery release on any of these cloud platforms or a local system using Docker container. InterSystems does not provide maintenance releases for continuous delivery releases, but instead fixes issues in subsequent continuous delivery releases.

The initial major extended maintenance release is provided on all [InterSystems IRIS Supported Platforms](#), including UNIX, Windows, the cloud platforms, and the Docker container. Following maintenance releases are provided on all server and

cloud platforms in the *InterSystems IRIS Supported Platforms*, but are not provided on the Docker container. If you are on a Docker container, you can upgrade to a continuous delivery release.

If your application runs on a non-container platform, you can only use an extended maintenance release for that application but can consider using the continuous delivery releases for:

- Evaluating new features and testing your custom code — this will reduce your upgrade costs when you upgrade to the next major release.
- Using it for new projects that can be deployed in the cloud or in local containers.

In addition to providing fully-supported releases, InterSystems provides access to prerelease software for developers who want to get an early look at new features.

## 1.2 SQL Enhancements

As with every release, further enhancements were made to the InterSystems IRIS SQL engine to further improve the performance of your SQL workload. Specifically, this release includes improvements to our parallelization engine that enable more types of queries and DML to be parallelized (automatically) and make more efficient use of CPU capacity. Also, sharded queries can now use implicit joins using `->` or `*=` syntax. Please also observe the general recommendations on upgrading SQL-based applications described in “[Frozen Plans for SQL Queries](#)” in General Upgrade Information.

## 1.3 Infrastructure and Cloud Deployment Improvements

This release contains improvements to the infrastructure and cloud deployment, including the following:

- Tencent Cloud Support — InterSystems Cloud Manager (ICM) now provides end-to-end cloud provisioning and deployment for applications based on InterSystems IRIS and running on Tencent Cloud.
- Support for Docker named volumes in addition to bind mounts.
- InterSystems Cloud Manager (ICM) support for elastic scaling (scale out DATA nodes, and scale in/out COMPUTE nodes).

## 1.4 Native API for Java and .NET Enhancements

The IRIS Native API, which allows you to access InterSystems IRIS data using globals, has been extended to include `$LIST`, so that developers can easily iterate through their data structures without needing to parse them in detail. This supports scenarios such as developing Java and .NET applications that access existing global structures from existing applications, as well as simplifying development and speeding performance.

## 1.5 New Automatic Configuration Customization

A new InterSystems IRIS configuration feature enables customization of the configuration parameter file (CPF) of an InterSystems IRIS instance prior to startup, upon which the custom configuration is automatically implemented. This feature

greatly simplifies automation and supports the use of configuration management tools such as Kubernetes with InterSystems IRIS., and is also included in ICM in this version. Automatic configuration customization is an important new capability that will be expanded in future versions.

## 1.6 Other Enhancements and Efficiency Improvements

In each release, InterSystems makes many efficiency improvements and minor enhancements. In this release these improvements include:

- Source Control for Productions — source control hooks have been added to allow check-in and check-out of a production as an entity, simplifying change tracking and configuration management.
- Whitelists to Support Penetration Testing — customers performing their own security penetration testing can reduce or eliminate false positives related to CSP, Zen, and REST.



# 2

## General Upgrade Information

This section provides information on upgrading from earlier versions. The ultimate goal of InterSystems is to have a release which can be installed with no, or little, effect on the applications it supports.

### 2.1 Important Considerations

#### 2.1.1 Compatibility

The goal of each release is a smooth upgrade to new and improved functionality with no changes required to existing applications. However, because of error corrections, significant enhancements, or changes to applicable standards, this goal cannot always be met. In this case, InterSystems discloses all identified instances where changes to applications are necessary so that customers can gauge effort required to move to the new release.

You may, after reading the release-specific changes, conclude that none of them affect your application. Even in this case, regardless of how robustly designed and how well implemented your application is, there is no substitute for quality assurance test results that confirm your judgement and demonstrate the application remains unaffected by the upgrade.

**Important:** InterSystems recommends that each application be thoroughly tested in the upgraded environment **before** it is deployed to customers and begins processing live data.

#### 2.1.2 Preview Release

Toward the end of development for each release, InterSystems may make available a preview version of the product to its customers. Notifications of the preview are published on the website and on public blogs. The purpose of this is two-fold:

- It provides an early opportunity for customers to determine how the changes and enhancement in the release affect existing applications, to report issues found, and verify those issues have been resolved.
- It also provides early exposure of new features. Customers have a chance to try out the proposed ideas and give feedback on the usefulness of this feature for their business area.

InterSystems strongly encourages customers to plan on obtaining a preview release and to test their application against it.

**Important:** InterSystems does not support upgrading from a preview version.

## Unresponsive Systems

One of the goals for preview release is to expose the new release to real-world operating challenges to assure its reliability. Therefore, it is possible, although unlikely, that an unanticipated sequence of events may render InterSystems IRIS unresponsive. In this situation, it is extremely important to gather system diagnostic information while in the hung state for InterSystems to analyze. Should an instance of InterSystems IRIS become unresponsive,

- Log in as an administrator
- In a terminal window, run the [IRISHung script](#) in the directory, <install-dir>/bin.

The scripts corresponding to supported systems are:

- Windows: IRISHung.cmd
- UNIX®, Linux, AIX, and so on: IRISHung.sh

- Send the resulting output file to the InterSystems [Worldwide Response Center](#). You can email the file to support@intersystems.com, open a new problem using the WRC Online, or call the Center directly for additional assistance.

## 2.2 Upgrade Specifics

This section contains specific instructions applicable to this transition.

### 2.2.1 Upgrading Containers

Because a containerized application is isolated from the host environment, it does not write persistent data; whatever it writes inside the container is lost when the container is removed and replaced by a new container. Therefore, an important aspect of containerized application deployment is arranging for data to be stored outside of the container and made available to other and future containers.

The durable %SYS features enables persistent storage of instance-specific data — such as user definitions, audit records, and the log, journal, and WIJ files — when InterSystems IRIS is run in a container, allowing a single instance to run sequentially in multiple containers over time. For example, if you run an InterSystems IRIS container using durable %SYS, you can upgrade the instance by stopping the original container and running a new one that uses the instance-specific data created by the old one. For information about upgrading, see [Upgrading InterSystems IRIS Containers](#); for detailed information on durable %SYS, see [Durable %SYS for Persistent Instance Data](#).

**Important:** In this release, the distribution container has a nonroot default user. This improves the security of your container. If you are using a durable %SYS from a 2019.2 or earlier instance with this 2019.4 release, you need to change some file ownerships in the host's durable directory before running InterSystems IRIS 2019.4. Please contact your InterSystems sales engineer or the InterSystems [Worldwide Response Center](#) for instructions on changing the file ownerships. If you do not make these changes, the container will encounter an error starting InterSystems IRIS.

### 2.2.2 Classes

InterSystems recommends that customers recompile all their classes contained in each namespace. This will assure that:

- Subclasses derived from the InterSystems product library will see improved product behavior if a method call results in executing code in its superclass(es).
- All embedded SQL will use the latest versions of the SQL infrastructure.

- All projections involved in language bindings will be updated.
- All generated routines and classes will be updated.

### 2.2.2.1 Class compiler version utility

To assist customers in determining which class compiler version a class or classes in a namespace have been compiled with, InterSystems provides two assists

- Method – \$System.OBJ.CompileInfoClass(<classname>)

This method returns the version of the class compiler used to compile this <classname> and the datetime the class was compiled

- Query – \$System.OBJ.CompileInfo(<sortby>)

This query generates a report for the current namespace that includes all classes, the version of the compiler used to compile each one, and the datetime each was compiled. The first argument <sortby> may have the following values:

- 0 – the time the class was compiled
- 1 – the class name
- 2 – the version of InterSystems IRIS the class was compiled in

### 2.2.3 Routines

ObjectScript routines do not need to be recompiled after upgrade with the following exception:

- Routines containing embedded SQL must be recompiled.

### 2.2.4 Cached Queries

Cached queries are always purged during upgrade. They are recompiled and cached as needed.

### 2.2.5 Web Services and SOAP

It is not necessary to re-import any Web Service Definition (WSDL) files.

### 2.2.6 Frozen Plans for SQL Queries

When you upgrade InterSystems IRIS to a new major version, existing Query Plans are automatically frozen. This ensures that a major software upgrade will never degrade the performance of an existing query. For performance-critical queries, you should test if you can achieve improved performance. For details, see “[Software Version Upgrade Automatically Freezes Plans](#)”

in the *InterSystems SQL Optimization Guide*.



# 3

## Upgrade Compatibility Checklist for InterSystems IRIS 2019.4

The purpose of this chapter is to highlight those features of the 2019.4 release of InterSystems IRIS® data platform that, because of their difference in this version, affect the administration, operation, or development activities of existing systems.

This document addresses only the differences between the 2019.3 and 2019.4 versions of InterSystems IRIS. Customers upgrading their applications from earlier releases are strongly urged to read the upgrade checklist for the intervening versions as well:

- [Upgrade Compatibility Checklist for InterSystems IRIS 2019.3](#)
- [Upgrade Compatibility Checklist for InterSystems IRIS 2019.2](#)
- [Upgrade Compatibility Checklist for InterSystems IRIS 2019.1](#)

### 3.1 Administrators

This section highlights information of interest to those who are familiar with administering prior versions of InterSystems IRIS and wish to learn what is new or different in this area for version 2019.4. The items listed here are brief descriptions. More complete descriptions are in the following section.

#### 3.1.1 Improved Container with Nonroot Default User Requires Installation Change

In this release, the distribution container has a nonroot default user. This improves the security of your container. However, if you are using a durable %SYS from an InterSystems IRIS 2019.2 (or earlier) instance, you need to change some file ownerships in the host's durable directory before running InterSystems IRIS 2019.4. Please contact your InterSystems sales engineer or the InterSystems [Worldwide Response Center](#) for instructions on changing the file ownerships. If you do not make these changes, the container will encounter an error starting InterSystems IRIS.

For information about upgrading, see [Upgrading InterSystems IRIS Containers](#), and for detailed information on durable %SYS, see [Durable %SYS for Persistent Instance Data](#).

## 3.1.2 Changes for Dejournaling

There are several dejournaling changes, which have some impact on system administration:

- On systems configured with a database cache larger than 128GB, a dejournaling task (which applies database updates in the journal to databases during a journal restore or when the backup failover member of a mirror is synchronized with the primary) may employ more updater processes than in previous versions, and therefore use more system resources. If your instance is close to resource limits, this may cause problems.
- Previous versions of Caché or InterSystems IRIS allowed fewer databases to be mounted at one time than the current version. Any journal files created after an instance has passed the old limit are therefore incompatible with previous versions of Caché or InterSystems IRIS. For example, they could not be used to run a journal restore after restoring databases from backup to an instance of a previous version. However, journal files created by previous versions are compatible with this version.
- If dejournaling is used as part of recovery after a system crash, for example in journal restore, it is possible that inter-global inconsistency could occur in a dejournalized (restored) database, although the data within a global is still guaranteed to be logically consistent.

## 3.2 Developers

This section contains information of interest to those who have designed, developed and maintained applications running on prior versions of InterSystems IRIS. Since developers typically administer development system, developers should also read the previous section for administrators.

The items listed here are brief descriptions. In most cases, more complete descriptions are available elsewhere in the documentation.

### 3.2.1 Custom Code Cannot Assume Root Privileges Running in Docker Containers

In previous releases, the Docker container had a default root user, but in this release the container has a nonroot default user (irisuser, UID 52773) to improve container security. Consequently, any custom code that assumes that the default user has root privileges running in Docker containers will fail in this release. Processes which connect to an IRIS container (such as via JDBC) should not be affected unless they execute code that makes assumptions about rootlike privileges when dealing with files. Note that container-root already lacks, by default, many of the kernel capabilities usually held by host-root in noncontainer systems.

This change only applies to Docker images provided by InterSystems. If you create a Docker image using a noncontainer install, you can control the default user.

### 3.2.2 Changes in SQL TRUNCATE TABLE Transactions

TRUNCATE TABLE no longer initiates transactions when executed in AUTOCOMMIT\_ON or AUTOCOMMIT\_OFF mode. If you need the ability to rollback a truncate table command, you must start your own transaction prior to executing the truncate table.

### 3.2.3 TSQL Parser Changes

Prior to this change, if you had defined:

---

`^%SYS("tsql","SET","ANSI_NULLS")="OFF"`

or

`^%SYS("tsql","SET","QUOTED_IDENTIFIER")="OFF"`

The TSQL parser would have behaved as if these settings were ON. Now that this has been corrected, if your code was really relying on either of these settings being ON, and not they are OFF because of the `^%SYS("tsql","SET","ANSI_NULLS")` or `^%SYS("tsql","SET","QUOTED_IDENTIFIER")` global setting, the run-time behavior of your procedure may be different.

### 3.2.4 Language Bindings Changes

Custom code using construct like `createIRISList()` or `IRIS.createIRISList()` should be changed to `IRISList.createIRISList()`.

Behavior of `DBList::toString` has been modified. Before this change both 01 and 02 01 was displayed as "null", now 02 01 displayed as "empty". Since `toString` is intended as a debugging tool, this change is unlikely to impact custom code.

### 3.2.5 Changes to TCP /BINDTO Behavior

If your application uses TCP device with `/BINDTO` option for outgoing connection, you may find the `OPEN` will fail if the IP does not exist in local system.

### 3.2.6 Changes to %SYS.PhoneProvider

In this release, you cannot delete a `%SYS.PhoneProvider` if it is in use. Some deleted operations that previously succeeded will now fail. This is an intentional feature but could cause some unexpected behavior in unusual cases.

### 3.2.7 Changes in Display of Host Status in Interoperability Productions

The meaning of the running status display in the Production Configuration page has changed because the changes to support source control for productions has involved a slightly different possible state. The status in the production configuration diagram for items that are enabled but not running, either because the item has a schedule and is currently stopped or if the item has been temporarily stopped, has changed. These items now have the status stopped which means that the item dot is shown light green as per the existing legend: Not running, enabled. Therefore 'Not running' can apply if the production is stopped or if the item is not running but the production is running.

### 3.2.8 Changes in X12 Acknowledgement Documents in Interoperability Productions

In previous releases, X12 acknowledgement documents had many wrong field values in them. This release corrects these values, but if custom code is expecting those incorrect values, you need to make modifications to accept the corrected values or implement `OnConstructReply()` in your Business Service instead.

### 3.2.9 Changes in syncresponses Array in BPLs for Interoperability Productions

The syncresponses array was not being correctly populated with the callresponses. Instead the process's response object was being used. It is unlikely that custom code will rely on syncresponses containing the response object. However, if this is the case then reworking of the code will be necessary.

### **3.2.10 Changes to EnsLib.EDI.Segment Class for Interoperability Productions**

If you have subclassed EnsLib.EDI.Segment and are making use of the RawContent property (this would require creating the getter method RawContentGet()), then you now have to add RawContent to the list of properties in your custom class since it would no longer be inheriting this property from EnsLib.EDI.Segment.

### **3.2.11 Changes in HIPAA Schema for Interoperability Productions**

This change fixes the name of the ISA:11 field for HIPAA\_5010. The old name of Interchange Control Standards Identifier was correct for older versions, but the purpose of this field changed in version 403 to now be the Repetition Separator. If you have used the HIPAA\_5010 schema and are getting or setting this field using the property path, ISA:InterchangeControlStandardsIdentifier, you will need to switch to using ISA:RepetitionSeparator or ISA:11. This change is relevant for IRIS for Health and Health Connect.

### **3.2.12 Changes in Handling Unexpected HL7 Names in Interoperability Productions**

In order to see information about unexpected HL7 Segment names you must now set ^Ens.Debug("TraceCat","parse") to 1 - this replaces possible event log warnings. Message validation can still be used to enforce schema segment names.

# 4

## Known Issues and Notes

This chapter describes issues in InterSystems IRIS® data platform 2019.4 that you should be aware of and other notes related to the release.

### 4.1 Cloud Manager Support for SUSE Linux

With InterSystems IRIS 2019.4, The InterSystems Cloud Manager (ICM) for SUSE Linux only supports SUSE 12-SP3.

