InterSystems™
IRIS Data Platform

Release Notes

Version 2019.2
2019-09-12

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

Release Notes
InterSystems IRIS Data Platform Version 2019.2 2019-09-12

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

About This Book

This book describes the major features that have been added to InterSystems IRIS Data Platform™ 2019.2, as well as information needed to update custom code from InterSystems IRIS™ 2019.1.

It contains the following sections:

- New and Enhanced Features for InterSystems IRIS 2019.2
- General Upgrade Information
- Compatibility Checklist for InterSystems IRIS 2019.2 — 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.

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™ 2019.2

This document describes the 2019.2 release of the InterSystems IRIS Data Platform™. The following sections describes this release and its new capabilities and enhancements:

- First Continuous Delivery release of InterSystems IRIS
- API Management
- Client Language Support including:
  - InterSystems IRIS Native API for Python
  - InterSystems IRIS Native API for Node.js
  - Relational access for Node.js
- Sharding enhancements including simplified architecture and flexible sharded schema design
- SQL enhancements
- Analytics enhancements including:
  - Czech language model
  - PowerBI connector
  - Pivot Table Preview
- New look in the Management Portal
- Cloud Manager Enhancements

For a list of new features in the previous version, see the InterSystems IRIS 2019.1 Release Notes and Upgrade Checklist.

1.1 First Continuous Delivery Release of InterSystems IRIS

InterSystems IRIS 2019.2 is the first 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-suppported releases, InterSystems provides access to preview software for developers who want to get an early look at new features.

1.2 API Management

This release includes two new API Management features:

- InterSystems API Manager
- Open API/Swagger Specification-First REST Development

1.2.1 InterSystems API Manager

The API Manager enables you to monitor and control traffic to and from your web-based APIs. If you are building service-oriented application layers, you are very likely to find the number of APIs you are using quickly rise. The more distributed your environment the more critical it becomes to properly govern and monitor your API traffic. The API Manager enables you easily route all your traffic through a centralized gateway and forward API request to appropriate target nodes. This enables you to:

- Monitor all your API traffic in a central spot.
- Plan, document, and update the list of APIs you are using and the servers that provide them.
- Identify issues before they become critical.
- Control API traffic by throttling throughput, configuring allowed payload sizes, whitelist and blacklist IP addresses and domains, and quickly taking an endpoint into maintenance mode.
- Onboard internal and external developers by providing interactive API documentation through a dedicated and customizable developer portal.

- Secure your API's in a central place.

The API Manager is interoperable, reliant, intuitive, and scalable. You can perform all configuration using a simple web-based user interface, but can also configure the API Manager using API calls, which makes it easy to perform remote deployments,

The API Manager is released in its own container. You can configure the API Manager as a cluster of multiple nodes, but even a single node can handle the load of multiple tens of thousands of requests per second.

For more information, see Using the InterSystems API Manager.

1.2.2 Open API/Swagger Specification-First REST Development

This release enhances the API Management service so that it can generate the ObjectScript code for REST services from OpenAPI 2.0 specifications. This generated code handles the incoming REST call and you only have to write custom code to perform the specific function performed by the service. If you are implementing a service that is already defined in an OpenAPI 2.0 specification, your work is significantly reduced. Even if there is no existing OpenAPI 2.0 specification, it is much easier to create a new specification than to write the custom code required to define the REST API and the specification also provides documentation and aids anyone developing client code for the service. For details see Creating REST Services.

1.3 InterSystems IRIS Native API for Python

This release introduces the Native API for Python, which is a lightweight Python interface to the native multidimensional storage data structures that underlie the InterSystems IRIS™ object and SQL interfaces. The Native API allows you to implement your own data structures by providing direct access to global arrays, the tree-based sparse arrays that form the basis of the multidimensional storage model. By providing direct access to global arrays, the Native API for Python allows you to define very efficient storage structures in Python and, consequently, very efficient applications. For details see Using the Native API for Python and Native API for Python Reference.

1.4 InterSystems IRIS Native API for Node.js

This release introduces the Native API for Node.js, which is a lightweight Node.js interface to the native multidimensional storage data structures that underlie the InterSystems IRIS™ object and SQL interfaces. The Native API allows you to implement your own data structures by providing direct access to global arrays, the tree-based sparse arrays that form the basis of the multidimensional storage model. By providing direct access to global arrays, the Native API for Node.js allows you to define very efficient storage structures in Node.js and, consequently, very efficient applications. For details see Using the Native API for Node.js and Native API for Node.js Reference.

1.5 Relational access for Node.js

This release provides ODBC access to InterSystems IRIS databases to Node.js developers

1.6 Sharding Enhancements — Simplified Architecture and Flexible Sharded Schema Design

This release introduces a simple and straightforward blueprint for sharded clusters, the node-level architecture, which can be configured through the new %SYSTEM.Cluster API. In this cluster architecture, we've implemented some best practices on laying out the different foundational elements introduced with sharding in the initial IRIS 2018.1 release that will make it significantly easier to deploy and expand your cluster. As the node-level architecture is essentially a smart way of leveraging the existing infrastructure, it is fully transparent to application code and does not require any changes to existing deployments. For details, see "Node-Level Architecture" and "Sharding APIs" in the *Scalability Guide* and see the %SYSTEM.Cluster class documentation in the *InterSystems Class Reference*.

This release introduces further improvements to how sharding supports designing your application's schema in the following ways:

- - You can now coshard any two sharded tables. Before, only tables with a common user-defined shard key could be cosharded (that is, explicitly defining the shard key for an Order and OrderLine table to be OrderID, the field on which they are joined). With this release, you can use COSHARD WITH syntax in DDL or the CoShardWith index keyword to coshard a new table with an existing table that has a system-assigned shard key. This significantly increases the flexibility for designing your application's schema, preserving the operational benefits of using system-assigned shard keys. For details, see "Create the Target Table" in the *Scalability Guide* and "Defining a Sharded Table" in *Using InterSystems SQL*.
- Where previously sharded schema design could only happen through DDL, you can now mark a persistent class (table) as sharded through its class definition, using the new "Sharded" class keyword. The class compiler has been extended to warn against using class definition features incompatible with sharding, such as customized storage definitions, at compile time. For details, see "Defining a Sharded Table by Creating a Persistent Class" in *Using InterSystems SQL*.
- Also starting with this release, you can use the object paradigm to interact with sharded classes. This means you can create new or open existing sharded class instances with %New() or %OpenId() methods and save them with %Save(). The sharding infrastructure will make sure new instances are appropriately distributed across the cluster, fully transparent to the application. Note that this object code will still be executed on the machine your client is connected to.

1.7 SQL Performance Enhancements

As with every release, InterSystems IRIS includes a number of enhancements to its SQL engine, based on advancements in the underlying software and continuous benchmarking against industry-standard and customer workloads. Customers may observe measurable increases in query throughput for high-load scenarios compared to the 2019.1 release and are encouraged to share their experiences with InterSystems in case there is an opportunity for extending our benchmarking to include specific new use cases. Please also observe the general recommendations on upgrading SQL-based applications described in Frozen Plans for SQL Queries in the *Upgrade Checklist*.

1.8 Czech Language Analytics Model

This release introduces support for the Czech language with InterSystems IRIS Natural Language Processing. The embedded NLP engine will now also identify concepts and their context for natural language text written in Czech as it does for the 10 other languages supported previously. Please refer to [GIKNOW] for more on leveraging NLP in your applications

1.9 PowerBI Connector

InterSystems customers can now use Microsoft Power BI to access tabular and cube data stored on InterSystems IRIS. This allows combining the data visualization capabilities offered by Power BI with the high-performance data management and querying capabilities offered by InterSystems IRIS. While the connector leverages ODBC, it will also allow customers to access InterSystems IRIS BI cubes directly from Power BI when connecting to InterSystems 2019.2 or above. The connector ships as part of Power BI starting with its April 2019 release. For details, see InterSystems IRIS Connector for Power BI .

1.10 Pivot Table Preview

This release contains the Analytics Pivot Table Preview, a new mode for the Analyzer that presents a representative pivot table based on a truncated data set. This will allow previewing a pivot table much more quickly than analyzing the complete result set. A **Show All** button is also presented when in Preview mode to indicate that the result set is not complete. Selecting the **Show All** button automatically turns off Preview mode.

1.11 New Look in the Management Portal

This release represents the beginning of a new, more modern look for the Management Portal. In this first phase, the menus and buttons have a new look but the functionality is unchanged. This new implementation provides the basis for future streamlining and improvements to the user interface.

1.12 InterSystems Cloud Manager Enhancements

In this release, the InterSystems Cloud Manager has the following enhancements:

- Elasticity improvements — Existing configurations can now be scaled, that is, reprovisioned and redeployed with more or fewer nodes. For details, see “Reprovisioning the Infrastructure” and “Redeploying Services” in the *InterSystems Cloud Manager Guide*.
- Containerless Mode — The following were previously restricted but can now be performed: deploying sharded configurations on Google Cloud Platform using containerless mode and deploying the Web Gateway on Ubuntu or SUSE nodes using containerless mode.

2

General Upgrade Information

This section provides information on upgrading from earlier versions. InterSystems' ultimate goal 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.

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 2019.2

The purpose of this chapter is to highlight those features of InterSystems IRIS™ 2019.2 that, because of differences in this version, affect the administration, operation, or development activities. If you are upgrading from InterSystems IRIS 2018.1, also see the 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 2019.2. There are no compatibility issues relevant to administrators.

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 Interoperability Production HTTP Generic Message Changes

The EnsLib.HTTP.GenericMessage class now explicitly defines XMLNAME and XMLTYPE parameters. Any custom subclasses of EnsLib.HTTP.GenericMessage need to account for XMLNAME parameter now being inherited and, if necessary, override it.