# InterSystems IRIS Migration Guide
for Caché and Ensemble Customers

Executive Summary

InterSystems IRIS Data Platform™ is a new product in the InterSystems portfolio that offers you a comprehensive platform for building critical applications that meet today's challenging requirements. InterSystems IRIS™ was built from the ground up to be Interoperable, Reliable, Intuitive, and Scalable – and goes beyond the capabilities offered by InterSystems Caché and Ensemble.

We encourage customers developing new applications to use InterSystems IRIS to take advantage of the powerful capabilities which it introduces. The new platform will also appeal to customers who have developed applications on Caché and Ensemble. InterSystems IRIS includes the vast majority of capabilities of both of those products and includes a number of libraries from them, but it is a new product built with an eye towards the future.

For customers who are migrating existing applications from Caché or Ensemble to InterSystems IRIS, the migration may require some changes to application code, configuration scripts, and other procedures, but will be relatively easy for the majority of cases.

Contents

Upcoming options for application conversion

InterSystems currently supports customers migrating from Caché/Ensemble to InterSystems IRIS as described in this document.

InterSystems has added an in-place conversion process to the InterSystems IRIS installer. If you have a large existing installed base, you may be interested in these facilities. It includes converting non-Unicode databases and allows for new installation of non-Unicode instances.

When this functionality is released, you can convert an existing Caché or Ensemble instance to InterSystems IRIS. This process will preserve all settings but will convert (transform) the instance into an InterSystems IRIS instance. The overall experience will be similar to upgrading a Caché or Ensemble instance to a new version. The InterSystems IRIS in-place Conversion Guide describes this in great detail.

Non-Unicode installation is available to enable existing Caché customers, not using Unicode, to use those databases directly with InterSystems IRIS. The non-Unicode instance of InterSystems IRIS offers the same character sets and collation tables as Caché.

This functionality will be available with the general release of InterSystems IRIS 2019.1.1. In-place conversions are supported for Caché and Ensemble versions 2017.1 and later.

IRIS Migration Initiative - Sequencing

Step	Migrations or Conversion from/to	Status
1	Migration to IRIS or IRIS for Health	Available today
2	Migration of TrakCare to IRIS for Health	Available today
3	In-place conversion to HealthShare Health Connect	Available today First version: InterSystems IRIS 2019.1.1
4	In-place conversion to IRIS	Available today First version: InterSystems IRIS 2019.1.1
5	In-place conversion to IRIS for Health	Available today First version: InterSystems IRIS 2020.1
6	In-place conversion to HealthShare	Available today First version: HealthShare 2019.2

Introduction

This document is meant to help existing Caché and Ensemble customers who are considering InterSystems IRIS.

InterSystems is encouraging all customers to move existing applications from Caché and Ensemble. InterSystems will continue to support Caché and Ensemble with new releases and is committed to the high standards of customer service through its Worldwide Response Center for those products.

If you want to add new capabilities from InterSystems IRIS into your existing applications as soon as possible, we will help you. As a first step, we advise you to review how your application can take advantage of the new capabilities and the timeframe in which you expect to need them.

You should review this document as well as the InterSystems IRIS documentation for the new capabilities you are interested in, and discuss your project with InterSystems.

You may also want to get hands-on experience with some of the new capabilities before starting your project. We provide a variety of ways to do this, including hands-on environments and experiences through the InterSystems IRIS content on our Online Learning portal and with the InterSystems IRIS Community and Express editions.

Migrating an existing Caché or Ensemble application can be simple and smooth. However, such migrations should be planned carefully, because there are a few areas that are not backward-compatible. You should review the elements that are different in InterSystems IRIS and identify whether any of them could impact your application.

Overview of Differences

At a high level, there are five categories of differences related to migrating existing applications:

A. **Platforms and browsers:** Some platforms and browsers supported with Caché are not supported by InterSystems IRIS.

B. **Naming**: InterSystems IRIS is a new product, and therefore several code and system artifacts have a name that differs from what you used in Caché or Ensemble.

C. **Application Development**: This section describes changes that might require changes in your applications. Most are simple renaming changes, but the SQL delimited identifiers might require some careful investigation and testing.

D. **Behavior**: InterSystems IRIS also includes a small number of behavioral differences. Most of these are new defaults, some of which had already been our recommendation for quite some time with Caché and Ensemble.

E. **Legacy modules**: Several legacy modules or self-contained features in Caché and Ensemble are not available with InterSystems IRIS (such as Zen Mojo) or have only been included to help with migration and will be phased out in a future version (such as DeepSee Visual Reports). If you are relying on such functionality please feel encouraged to contact InterSystems to discuss alternatives.

F. **Healthcare-specific features**: Healthcare-related features (such as HL7 support) that are available in Ensemble, are not part of the InterSystems IRIS Data Platform standard release. These are included with InterSystems IRIS for Health.

G. **Systems setting**: System settings are not compatible across Caché/Ensemble and InterSystems IRIS. The conversion process will transform system settings to the format.

Using new features: There are many new capabilities in InterSystems IRIS that support scenarios that were impossible to implement with Caché and Ensemble. Handling most of these new capabilities will involve developing new application code. Some of them, like sharding for horizontal scalability or container packaging, are transparent to application *code*, but exploiting their benefits to the fullest still involves specific system and application *deployment*. InterSystems recommends migrating to InterSystems IRIS in a broader modernization context to allow you to take full advantage of the new capabilities it offers.

Note: This document is current with InterSystems IRIS 2022.1, and will evolve with future releases.

Platforms and Browsers

Platforms

The list of supported server platforms with InterSystems IRIS is provided here. InterSystems IRIS development platform are not supported by the in-place conversion process.

Certain older platforms, which are supported with Caché and Ensemble, are not supported with InterSystems IRIS. These include 32-bit platforms; Windows versions before Windows 10,0 HP-UX, Solaris, AIX 6.1, Red Hat 6, and SUSE 11. Applications deployed on these platforms should be migrated to a supported operating system.

Browsers

The list of supported browsers is provided here.

Certain older browsers are not supported. When migrating to InterSystems IRIS, plan to move clients accessing affected interfaces to a recent browser.

Different Names

InterSystems IRIS is a new product and therefore several code and system artifacts have names that differ from what you used in Caché or Ensemble.

Differences in file and other artifact names may affect scripts, procedures, and documentation; differences related to application development will require updating and recompiling application code.

System

A number of core-executables and libraries have new names. For example, you will see an **irisdb.exe** file (and process) on Windows where you have seen **cache.exe** in the past. See the list below for differences that we expect to affect your code or scripts.

If you have any user code or scripts relying on these names, update them when you migrate to InterSystems IRIS.

1. Executables to start, stop, and connect to InterSystems IRIS have different names, as do install scripts:

Caché / Ensemble	InterSystems IRIS	Comment
/bin/ccontrol	**/bin/iris**	
/bin/csession	**/bin/iris session**	Please use /**bin/iris help** for details Note: On Windows the argument **session** is not yet supported, please use one of the other options (terminal, console, run). Note: In keeping with other command-line utilities, the operation of starting a session is now an option in the main script.
/bin/cstat	**/bin/irisstat**	
cinstall	**irisinstall**	
cinstall_silent	**irisinstall_silent**	

2. Key database artifacts have different names:

Caché / Ensemble	InterSystems IRIS	Comment
CACHE.DAT	**IRIS.DAT**	Per database
cache.lck	**iris.lck**	Per database
cache.use	**iris.use**	Per instance
CACHE.WIJ	**IRIS.WIJ**	Per instance

Note that, for migration purposes, you can simply rename an existing CACHE.DAT to IRIS.DAT and it will be picked up as usual. Of course, this will not automatically update any global, class, method, routine etc. names inside the .DAT file. These will still have to be updated manually, according to the overview provided in this chapter.

Note that for a number of scenarios where code elements have corresponding data points stored in reserved globals, such as an iKnow domain definition, this renaming of .DAT files will not work and exporting and importing the code (with any relevant alterations) is the only option. InterSystems recommends to using the upcoming in-place conversion mechanism, which will automatically perform the correct steps.

The ENSLIB Database will mount as read-only with InterSystems IRIS. InterSystems recommends keeping it as read-only for production installations, and suggest that dependencies in your application on writing to this database are resolved.

3. Key database names differ, as do their associated directories and resource names:

Caché / Ensemble	InterSystems IRIS	IRIS Resource
CACHELIB	**IRISLIB**	%DB_IRISLIB

CACHESYS	**IRISSYS**	%DB_IRISSYS
CACHETEMP	**IRISTEMP**	%DB_IRISTEMP
CACHEAUDIT	**IRISAUDIT**	%DB_IRISAUDIT
CACHE	**IRISLOCALDATA**	%DB_IRISLOCALDATA

4. On UNIX™, environment variables and accounts are different:

Caché / Ensemble	InterSystems IRIS	Comment
CACHESYS	**IRISSYS**	Environment variable
cacheusr	**irisusr***	Username and group for managing ownership

*Converting an existing Caché or Ensemble instance to InterSystems IRIS will continue to use cacheusr. That is ok and should not be changed. New IRIS installations will use irisusr.

5. The file names for certificate-related entities are different:

Caché / Ensemble	InterSystems IRIS	Comment
cache.cer	**iris.cer**	Certificate file
cache.crl	**iris.crl**	Certificate revocation list

6. The main log file name is different:

Caché / Ensemble	InterSystems IRIS	Comment
cconsole.log	**messages.log**	

7. The main configuration file name is different:

Caché / Ensemble	InterSystems IRIS	Comment
cache.cpf	**iris.cpf**	

8. CSP URL parameter (login token) names are different (though, to ease migration, InterSystems IRIS correctly handles parameters with names that include “Cache”):

Caché / Ensemble	InterSystems IRIS	Comment
CacheUserName	**IRISUserName**	
CachePassword	**IRISPassword**	
CacheLogin	**IRISLogin**	
CacheNoRedirect	**IRISNoRedirect**	
CacheOldPassword	**IRISOldPassword**	
CacheRepeatPassword	**IRISRepeatPassword**	
CacheLogout*	**IRISLogout**	
CacheLoginPage	**IRISLoginPage**	
CacheSecurityToken	**IRISSecurityToken**	
CacheSecurityCancel	**IRISSecurityCancel**	
CacheSecuritySubmit	**IRISSecuritySubmit**	

*CacheLogout is not supported on InterSystems IRIS, please use IRISLogout.

9. The default collation name is different:

Caché / Ensemble	InterSystems IRIS	Comment
Cache standard	**IRIS standard**	
Cache standard string	**IRIS standard string**	

10. The names of many DLLs and other files within the installation directory are different. As a general guideline, files beginning with **c** now begin with **iris**, and files beginning with **Cache** now begin with **IRIS**, for example:

Caché / Ensemble	InterSystems IRIS	Comment
CacheObjectScript.dll	**IRISObjectScript.dll**	
cconnect64.dll	**irisconnect64.dll**	
cctrl.dll	**irisctl.dll**	

11. The names of the following commonly used system monitoring and diagnostic tools are different:

Caché / Ensemble	InterSystems IRIS	Comment
^pButtons	**^SystemPerformance**	
^Buttons	**^SystemCheck**	

Application Development

To migrate from Caché and Ensemble to InterSystems IRIS, you will need to change and recompile application code. A few well-aimed search-and-replace operations should speed and simplify the process. A very limited number of public interfaces are different and appear below by language type.

SQL Development

Delimited Identifiers

InterSystems IRIS more tightly enforces rules for delimited identifiers:

In SQL parsing, when support for delimited identifiers is on (the default):

- if "" (double quotes, no spaces) is specified in the statement, an SQLCODE=-1 error is returned.
- "" is not a valid identifier in InterSystems IRIS SQL.

In Caché/Ensemble, "" would be recognized as an empty string, but this is incorrect, and in InterSystems IRIS an error is now returned.
To specify an empty string in an SQL statement, use ''.

In addition, InterSystems IRIS does not support the following delimited identifiers.

- An identifier that contains the "^", ".", or "," character
- An identifier that contains the character sequence "=>"
- An identifier that begins with the "*" character
- An identifier equal to "%VID", in any case.

An SQLCODE=-1 error is returned if you attempt to use a delimited identifier that contains something that is not supported.

With InterSystems IRIS, the names for a stored procedure and a class query cannot be the same. With Caché and Ensemble it is possible to have both with the same name.

Operator Precedence

With InterSystems IRIS, the default operator Precedence changed to the ANSI behavior. For example: “SELECT 1+1*10” will now return 11 and not 20. The in-place conversion will preserve the setting of the Caché/Ensemble instance. You can configure either type of SQL operator precedence system-wide using the $SYSTEM.SQL.SetANSIPrecedence() method.

ObjectScript Development

The following reserved global name patterns are different:

Caché / Ensemble	InterSystems IRIS	Comment
^ISC.*	**^IRIS.***	Internal, hidden by default. ^ISC.* is also reserved
^%ISC.*	**^%IRIS.***	Internal, hidden by default. ^%ISC.* is also reserved
^CacheTemp*	**<none>**	You are free to define a preferred global name pattern for temporary globals and map it to IRISTEMP. For compatibility purposes, ^CacheTemp* will continue to be mapped to IRISTEMP automatically.

^CacheAudit*	**^IRIS.Audit***	Mapped automatically to the IRISAUDIT database.
^||%CacheStream	**^||%IRIS.Stream**	^CacheStream is still used for permanent storage for compatibility reasons.
^SQL.Shell	**^IRIS.SQL.Shell**	Now internal.
^CacheMsg	**^IRIS.Msg**	Now internal.
^CacheMsgNames	**^IRIS.MsgNames**	Now internal.
$ETRAP	**<deprecated>**	$ETRAP is deprecated for new development with InterSystems. Please use Try/Catch instead.

1. Classes, methods, or triggers in ObjectScript should use the new implementation language “objectscript” rather than the old “cache” (which is still supported for compatibility). CSP applications should not be changed to use “objectscript”.

2. The $$$CacheError macro in Caché is named $$$ObjectScriptError in InterSystems IRIS.

3. Certain key classes have different names:

Caché / Ensemble	InterSystems IRIS	Comment
%CacheString	**%RawString**	Former still silently accepted
%CacheCollection %CacheDynamic %CacheEmbedded %CacheFileStream %CacheIndex %CacheLiteral %CacheObject %CachePopulate %CacheRegObj %CacheRelationship %CacheStream %CacheStreamEmbedded %CacheStreamLegacy %CacheTrigger	**%Compiler.Type.Collection** **%Compiler.Type.Dynamic** **%Compiler.Type.Embedded** **%Compiler.Type.FileStream** **%Compiler.Type.Index** **%Compiler.Type.Literal** **%Compiler.Type.Object** **%Compiler.Type.Populate** **%Compiler.Type.RegisteredObject** **%Compiler.Type.Relationship** **%Compiler.Type.Stream** **%Compiler.Type.StreamEmbedded** **%Compiler.Type.StreamLegacy** **%Compiler.Type.Trigger**	Note that, in many instances where application code refers to %CacheObject or %CacheLiteral, they actually mean to refer to %ObjectHandle and %CacheString (now %RawString), respectively.
%CacheStorage %CacheSQLStorage %CacheSerialState %Compiler.Storage.Generator.Cache %Compiler.Storage.Serial %Compiler.Storage.CacheSerial %Compiler.Storage.CacheSQL %Compiler.Storage.Extent %Compiler.Storage.CacheExtent %Compiler.Storage.Cache	**%Storage.Persistent** **%Storage.SQL** **%Storage.Serial** **%Compiler.Storage.Generator.Persistent** **%Compiler.Storage.CustomSerial** **%Compiler.Storage.Serial** **%Compiler.Storage.SQL** **%Compiler.Storage.CustomExtent** **%Compiler.Storage.Extent** **%Compiler.Storage.Persistent**	Old storage class names are converted automatically to new names on class save and in UDL import. Subclasses of classes with the old names require manual updates. Studio uses the correct classes.
%Dictionary.CacheClassname %Dictionary.CacheIdentifier	**%Dictionary.Classname** **%Dictionary.Identifier**	Now deprecated.

4. While the XML format for code exported from InterSystems IRIS is largely the same, the generator and version attributes in the header have changed. Importing code exported from Caché or Ensemble instance is transparently supported. To import code exported from InterSystems IRIS into a Caché or Ensemble instance (e.g. in order to maintain a single code base), first manually update the following attributes in the export fields:

   5.
      ```
      <?xml version="1.0" encoding="UTF-8"?>
      <Export generator="IRIS" version="26">
      <Class name="…">
      …
      ```

   Becomes:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <Export generator="Cache" version="25">
   <Class name="…">
   …
   ```

Java and JDBC Development

1. The main package name for Java is different:

Caché / Ensemble	InterSystems IRIS	Comment
com.intersys.*	**com.intersystems.***	

2. The main JAR files have different names. Applications leveraging Java should use the file names shown below (as found in \dev\java\lib\JDK1**):

Caché / Ensemble	InterSystems IRIS	Comment
cache-extreme-2.0.0.jar	**intersystems-xep-3.0.0.jar**	
cache-gateway-2.0.0.jar	**intersystems-gateway-3.0.0.jar**	
cache-jdbc-2.0.0.jar	**intersystems-jdbc-3.0.0.jar**	
<none> *	**intersystems-utils-3.0.0.jar**	
cache-db-2.0.0.jar	**<deleted>**	Java Binding discontinued

* The intersystems-utils-3.0.0.jar file has classes (XSLT gateway, QRCode) found in other JAR files in Caché.

3. Caché classes with "Cache" in their name have corresponding classes in InterSystems IRIS with "IRIS" in their name. For example, **com.intersys.jdbc.CacheDriver** has a corresponding **com.intersystems.jdbc.IRISDriver**.

4. JDBC URLs to an InterSystems IRIS instance have the form "**jdbc:IRIS://*host*:*port*/*namespace***"

5. The following methods in **com.intersystems.jdbc.IRISConnection** have different names:

Caché / Ensemble	InterSystems IRIS	Comment
get/setCacheJobID()	**get/setIRISJobID()**	
getVList(…)	**getListWriter(…)**	
recycleVList()	**recycleListWriter()**	

6. The startTransaction() method is not available from **com.intersystems.xep.EventPersister**.

.NET development

1. The file names associated with the .NET Gateway (sometimes referred to as the Object Gateway for .NET) are different. The following files are found in directories below **\dev\dotnet\bin\**:

Caché / Ensemble	InterSystems IRIS	Comment
InterSystems.Data.CacheClient.dll	**InterSystems.Data.IRISClient.dll**	
InterSystems.CacheExtreme.dll	**InterSystems.Data.XEP.dll**	
DotNetGatewaySS.exe	**InterSystems.Data.Gateway.exe**	
DotNetGatewaySS64.exe	**InterSystems.Data.Gateway64.exe**	

For more information about the .NET Gateway, see here.

2. All InterSystems ADO code is now declared in **InterSystems.Data.IRISClient**, and bindings code is declared in both **InterSystems.Data.IRISClient** and **InterSystems.Data.IRISTypes**. These replace InterSystems.Data.CacheClient and InterSystems.Data.CacheTypes respectively.

   The ADO class names are also different, which will require code updates. In general, all references to "Cache" now have been replaced by "IRIS"; this list summarizes the most important public classes that have different names:

Caché / Ensemble	InterSystems IRIS	Comment
CacheADOConnection	**IRISADOConnection**	
CacheConnection	**IRISConnection**	
CacheCommand	**IRISCommand**	
CacheCommandBuilder	**IRISCommandBuilder**	
CacheDataAdapter	**IRISDataAdapter**	
CacheDataReader	**IRISDataReader**	
CacheParameter	**IRISParameter**	
CacheTransaction	**IRISTransaction**	
Cache*	**IRIS***	

You will need to regenerate proxy classes to pick up the corresponding differences.

3. Applications leveraging the Entity Framework Visual Studio Tools should use the newly named VS tools file, InterSystems.Data.VSTools.

Other Application Development Name Differences

1. **ODBC Libraries**

   Applications leveraging ODBC with scripts or code containing explicit references to the ODBC libraries should be updated for the following name differences:

Caché / Ensemble	InterSystems IRIS	Comment
cgate.*	**odbcgateway.***	
CacheODBC.dll	**IrisODBC.dll**	
libcacheodbc*.so	**libirisodbc*.so**	

2. **New Dialect for Hibernate**

   Hibernate-based applications should now leverage the new **InterSystemsIRISDialect** dialect.

3. **Callin Functionality**

   The Callin functionality in Caché has a corresponding facility in the new product. Any items using this should be renamed from Cache* to IRIS*; this includes datatypes, constants, flags, error codes, and so on. Existing programs must be recompiled and linked before use with InterSystems IRIS, but no source changes should be required.
   The C header files needed for calling into InterSystems IRIS (iris-cdzf.h and iris-callin.h) are now located in the /dev/iris-callin/include/ subfolder (previously in /dev/Cache/callin). The old equivalents of these files are retained for compatibility.

Atelier

Atelier plugin names are different:

Caché / Ensemble	InterSystems IRIS	comment
com.intersys.eclipse.*	**com.intersystems.eclipse.***	

Analytics

The menu structure of the Management Portal differs, due to the different names of some embedded technologies:

Caché / Ensemble	InterSystems IRIS	Comment
DeepSee	**InterSystems IRIS Business Intelligence**	
iKnow	**InterSystems IRIS Natural Language Processing**	
iFind	**InterSystems IRIS SQL Search**	

As a result, some of the menu options are different:

Caché / Ensemble	InterSystems IRIS	Comment
DeepSee	**Analytics**	Top-level menu option
iKnow	**Analytics / NLP**	

Note that these naming differences have no impact on class or package names, so APIs for them are still available in the %DeepSee, %iKnow, and %iFind packages, respectively.

Different Packaging or Behavior

There are a number of areas in which InterSystems IRIS has different packaging, defaults, or behavior than Caché or Ensemble. If your application relies on any of the items listed in this section, you should discuss this with InterSystems and review these areas of your application when you migrate to InterSystems IRIS.

Documentation and Samples

The DOCBOOK namespace and database are not installed with InterSystems IRIS. The latest version of the documentation is always available at https://docs.intersystems.com/.

The SAMPLES and ENSDEMO namespaces and databases are not installed with InterSystems IRIS. Curated samples are available through our GitHub repository: https://github.com/intersystems/?q=samples.

Class reference documentation (Documatic) is available with InterSystems IRIS in the same fashion as with Caché.

Many Online Learning resources complementing the product documentation can be found at http://learning.intersystems.com/.

If you have a governance process that relies on locally available documentation, you will need to modify that process.

System Configuration

Item	Description and Recommendation
Default port numbers	The installing InterSystems IRIS installer will use the first available port starting with **(1972) 51773** for the SuperServer, the first available port starting with **52773** for the web server and first available port starting with **53773** for the JDBC Gateway.
License server port	InterSystems IRIS uses port **4002**; Caché uses port 4001.
Consolidation of web-related services and resources	The %Service_CSP, and %Service_MSMActivate services and associated resources are consolidated into **%Service_WebGateway**.
$zversion output	Applications that parse the $ZV string may need to be adjusted. InterSystems recommends that you use the methods of $System.Version instead, because these provide a future-proof API.
Long strings	Long strings are always enabled and can no longer be disabled.
Global buffers	4K global buffer sizes are not allowed and superseded by 8K, 16K, 32K, and 64K buffers. There is no impact to applications.

CPF File Differences

Some of the main configuration file entries are different or have different defaults:

Setting in cache.cpf	Difference in iris.cpf	Comment
[LicenseServers] port	**New default: 4002**	
[Startup] DBSizesAllowed	**Does not support 4K**	
[Startup] MaxCacheTempSizeAtStart	**Now MaxLocalDataSizeAtStart**	
[config] globals	**Does not support 4K**	
[Map.*] CacheMsg	**IRISMsg**	For Ensemble
[Map.*] CacheMsgNames	**IRISMsgNames**	For Ensemble
[Map.*] Global_CacheMsg	**No parameter**	For Ensemble
[Map.*] Global_CacheMsgNames	**No parameter**	For Ensemble
[Map.*] Global_EnsHL7	**No parameter**	For Ensemble
[Miscellaneous] EnableLongStrings	**No parameter**	
[SQL] JDBCGatewayAddress	**No parameter**	
[SQL] Prefix	**No parameter**	

With InterSystems IRIS, the [SQL] parameter group is sorted alphabetically.

Interoperability

InterSystems IRIS does not provide, by default, a namespace with the ENSEMBLE namespace mappings. If your application uses interoperability features, create a namespace and when doing so, ensure the checkbox "**Make this a production-enabled namespace**".

With InterSystems IRIS, the CSP Gateway is now known as the Web Gateway. You can access it from the Management Portal through **System Administration** / **Configuration** / **Web Gateway Management**.

Legacy Facilities and Modules

A number of facilities and modules used with Caché and Ensemble are not available with InterSystems IRIS, or have only been included in this first release for compatibility purposes for your migrating to InterSystems IRIS and will be removed in a future version. These features are clearly marked as deprecated in code and documentation artifacts. *Deprecated* indicates that something is available for the purposes of migration but not meant for use long-term. Discontinued indicates that a former deprecated technology is no longer supported. The tables below show items in this category, along with our recommendation. Please contact InterSystems to discuss the best approach for your application.

System Configuration and Operation

Description	Recommendation
Shadowing	Shadowing is discontinued. Applications using shadowing should migrate to the corresponding capabilities available with mirroring, as described in the High Availability Guide.

Interoperability

Description	Recommendation
TCP Provider	The TCP Provider is not available. If your application depends on the TCPProvider package, please contact InterSystems for guidance.
EISRequestWizard and HL7SchemaWizard	The EISRequestWizard and HL7SchemaWizard are no longer available in InterSystems IRIS. If your application depends on these wizards, please contact InterSystems for guidance.

Application Development

A number of language bindings, APIs, and libraries supported with Caché and Ensemble are not available (deprecated) with InterSystems IRIS. The table below lists these and the recommended change for application developers.

Description	Recommendation

Globals API	Applications using the Globals API (aka GlobalsDB) should migrate to JDBC, XEP, or the new Native API.
Java Binding	Applications using the Java Binding should migrate to JDBC, XEP, or Hibernate.
Jalapeño	Applications using Jalapeño should migrate to JDBC or Hibernate.
Python Binding	Applications using the Python Binding should migrate to PyODBC.
Perl Binding	Applications using the Perl Binding should migrate to ODBC.
C++ Binding and Light C Binding	Applications using the C++ or Light C Binding should migrate to ODBC.
.NET Binding	Applications using the .NET binding should migrate to ADO.NET, XEP, or the Entity Framework. Note that this only refers to the .NET *Binding.* A compatibility library is available from the WRC download server. Please check here for details. This technology is not supported for new development. **Existing Applications will require a special license key.**
.NET 3.0 Support	Applications using .NET 3.0 should migrate to a supported version (2.0, 4.0, or 4.5).
Caché Direct (VISM)	Applications using Caché Direct should migrate to ODBC. Existing applications may continue to use this technology, but will require a special license key to enable Caché Direct. In addition, customers need to copy the entire folder <CE installation directory>\dev\cdirect and put in the same location for the InterSystems IRIS installation. InterSystems will – at some point - provide an archive with the contents of the cdirect directory as a separate download from the WRC download server. This technology is not supported for new development. **Existing Applications will require a special license key.**
WebLink	Applications using WebLink should migrate to a modern web application framework. Existing applications may continue to use it, but will require a special license key to enable this. This technology is not supported for new development. Using WebLink is controlled by the %Service_Weblink service. **Existing Applications will require a special license key.**
Caché DB Library	Applications using DB Library (CacheDBLib.lib and cachedblib.h) should migrate to ODBC.
ActiveX controls	Applications using Activate.dll, TL.dll, and TL64.dll. Existing applications may continue to use this technology. InterSystems IRIS ActiveX supports ADO.NET 3.0 and later. The location of the files has changed to: C:\Program Files (x86)\Common Files\InterSystems\IRIS
Zen Mojo	Zen Mojo and its JavaScript and CSS libraries are no longer shipped with InterSystems IRIS. For applications that depend on any of these client libraries, either manually include them when deploying your applications or

	manually install Zen Mojo. A compatibility library is available from the WRC download server. This technology is not supported for new development. **Existing Applications will require a special license key.**

If your application is heavily dependent on language bindings and migration as recommended is prohibitive, contact InterSystems. We can provide individual language bindings for the purpose of migration for a limited term, but strongly recommend removing any dependence on such language bindings longer term.

Web Development with Zen

Zen is no longer a central piece of our application development strategy. It is included with InterSystems IRIS for compatibility and ease of migration, but is not recommended for new applications. If you are using Zen in your application with Caché or Ensemble, it will work with InterSystems IRIS, but you should look at porting or re-implementing it now or in the near future.

Our focus going forward is on providing a fast and scalable data management platform rather than GUI libraries. We are concentrating on leveraging the appropriate technologies to connect to the back end (such as REST) and suggest using best-of-breed third-party technologies of your choice for web development (such as Angular, React, Ember, Vue, etc.[1]). See also these recent courses on our Online Learning platform on how to leverage third-party web frameworks with InterSystems IRIS.

Analytics

Description	Recommendation
DeepSee I	DeepSee I is not available. Applications using DeepSee I (aka SpeedMiner, in %BI.* packages) should migrate to InterSystems IRIS Business Intelligence (previously known as DeepSee II).
DeepSee Visual Reporting	DeepSee Visual Reporting is deprecated. Applications using DeepSee Visual Reporting will still work for the time being, but you should contact InterSystems for guidance.
Zen Reports	InterSystems will continue to support Zen Reports but currently has no plans for major enhancements to the technology. Zen Reports will continue to be available in InterSystems IRIS, Caché and Ensemble for compatibility with existing applications.

[1] As of Oct 2018; Listed products are informational only and not an endorsement

Miscellaneous

The FileMan wizard has been removed from the Management Portal menu.

Several older or otherwise deprecated packages meant for internal use are not available with InterSystems IRIS. Many of these were already hidden and/or marked as deprecated in earlier Caché and Ensemble releases. These packages are:

- %Dashboard
- %DataModel
- %DeepSee.SetBuilder
- %EM
- %HadoopGateway
- %Service
- %Standards
- %UMLS
- %XEP
- XDEV

Available with InterSystems IRIS for Health

If your application depends on healthcare-related packages (such as ASTM, DICOM, HL7, and ITK), you should use InterSystems IRIS for Health.

The specific packages are:

- EnsLib.DICOM.*
- EnsLib.HL7.*
- EnsLib.Agent.*
- EnsLib.ITK.*
- EnsLib.EDI.ASTM.*
- EnsDemo.HL7v3.*

Coexistence & Compatibility

This chapter provides guidance for existing Caché and Ensemble customers that for practical reasons will need InterSystems IRIS to coexist or interact with Caché or Ensemble instances. For example, a new module may be developed on InterSystems IRIS, while an older one is still on Caché.

Recommendations and other notes on Caché and Ensemble in this section also apply to products based on them, such as any version of HealthShare or TrakCare that is not yet based on InterSystems IRIS for Health.

a. Installing InterSystems IRIS alongside Caché & Ensemble

Use of Virtual Machines or Docker Containers might address concerns with coexistence issues (e.g. isolation). See the Installation Guide and InterSystems Cloud Manager Guide for more details.

Backups created with the Caché or Ensemble Online Backup utility are not compatible with InterSystems IRIS.

b. Compatibility

In general, connectivity components created by InterSystems are not compatible between the two product lines. Industry standard protocols typically are, but you should be within the range of the minimum supported platforms.

InterSystems understands that some customers have

- applications or modules that they need to run, maintain and support on either InterSystems IRIS or Caché/Ensemble (same code targeting both products, but run independently).
- large deployments of their application that can't easily move to InterSystems IRIS entirely, but want to use features from IRIS (applications that use both products in combinations)
- large installed bases that must transition smoothly (temporary co-existence for migration purposes)

Connecting an InterSystems IRIS instance to an instance of Caché or Ensemble through ECP, mirroring or shadowing (the latter deprecated on InterSystems IRIS) is not supported as of version 2018.1.1, but support for these mixed configurations is provided in later releases of InterSystems IRIS.

In order to enable you to plan ahead, the following table lists all technologies that can be used in configurations where Caché/Ensemble based applications also connect to InterSystems IRIS:

Technology	Minimum Caché/Ensemble Version	Minimum InterSystems IRIS Version
ECP[1]	2017.1	2018.1
Failover and DR Async Mirror Members	2017.1*	2019.1
Reporting Async Mirror Members	2017.1*	2019.1
Web Gateway[2]	2017.1	2018.1
Rest/Web Services	2017.1	2018.1
LDAP	2018.1.3	2019.1.1
InterSystems Studio	2017.1	2018.1

***Important: to support mixed mirror configurations between the two product lines, Caché/Ensemble should have a specific enhancement (SML2736). SML2736 enables a Caché/Ensemble instance to discover an InterSystems IRIS failover mirror. This change is available with Caché/Ensemble 2018.1.3. Please contact the InterSystems WRC for earlier versions. For details on the process of converting failover mirror configurations to InterSystems IRIS, please consult the Conversion Guide.**

[1] Fully bidirectional supported for data access (Globals); Routines and Objects over ECP are not supported.

[2] InterSystems recommends using the Web Gateway for all C/E and InterSystems IRIS installations for version 2017.1 and later.

c. Client Compatibility

Client Technologies

InterSystems IRIS xDBC drivers (JDBC, ODBC and ADO.net) are not compatible with InterSystems Caché and Ensemble servers. This means applications connecting to an InterSystems IRIS instance need the InterSystems IRIS drivers and those connecting to a Caché or Ensemble instance need the Caché or Ensemble drivers, each of them coming with the regular kit.

Other client technologies, such as XEP, Entity Framework and Node.JS are not compatible between products. Hibernate has a new dialect for use with InterSystems IRIS. See also the

Application Development section in chapter B of the Overview of Differences earlier in this document.

d. Technology Compatibility: .Net Binding

In Caché and Ensemble, the ADO.NET and .NET Bindings functionality are bundled together in the same DLL named InterSystems.Data.CacheClient.dll.

In the case of InterSystems IRIS these components are distributed in separate DLL's, where ADO.NET features are included in InterSystems.Data.IRISClient.dll and Bindings functionality is included in InterSystems.Data.Bindings.dll. Most applications utilizing the .NET Binding will probably use ADO.NET features as well, so the migration steps for the most important ADO.NET types are listed here.

The general rule for all renaming is that instances of "Cache" within type names have been replaced with "IRIS" (upper-case), for example:

InterSystems.Data.CacheClient -> InterSystems.Data.IRISClient

To migrate a Bindings application from Caché or Ensemble to InterSystems IRIS:

1) Remove references to InterSystems.Data.CacheClient.dll from the .NET project. Add references to InterSystems.Data.IRISClient.dll *and* InterSystems.Data.Bindings.dll. If an application needs only ADO.NET features it can include only InterSystems.Data.IRISClient.dll but applications using Bindings features must include both DLLs.

2) Update the namespaces included in using directives throughout the code (see the namespaces section below).

3) Update all references to types defined in the two InterSystems DLLs (see the types section below).

4) Regenerate all proxy classes so that they use the updated types.

5) Rebuild the application.

Namespaces

Caché / Ensemble	InterSystems IRIS
InterSystems.Data.CacheTypes	InterSystems.Data.IRISTypes
InterSystems.Data.CacheClient	InterSystems.Data.IRISClient
InterSystems.Data.CacheClient.ObjBind	InterSystems.Data.IRISClient.ObjBind
InterSystems.Data.CacheClient.ObjBind.MetaInfo	InterSystems.Data.IRISClient.ObjBind.MetaInfo
InterSystems.Data.CacheClient.ObjBind.GeneratorOptions	InterSystems.Data.IRISClient.ObjBind.GeneratorOptions

Types

Many type names have changed -- first the important ADO.NET and Bindings types will be listed, then all data types. For the data types a convenient approach is to regenerate the desired proxy classes and refactor according to the types returned. Bindings applications will not use all the types below (mostly those in the top four sections) but all public types are included here for reference.

Important ADO.NET Types that might be used in Bindings applications

Caché / Ensemble	InterSystems IRIS
CacheConnection	IRISADOConnection
CacheCommand	IRISCommand
CacheParameter	IRISParameter
CacheDataReader	IRISDataReader
CacheDataAdapter	IRISDataAdapter
CacheParameter	IRISParameter

Bindings Connection

Caché / Ensemble	InterSystems IRIS
CacheConnection	BindingsConnection

Server object and class types

Caché / Ensemble	InterSystems IRIS
CacheObject	IRISObject
CachePersistent	IRISPersistent
CacheSerialObject	IRISObject
CacheRelationshipObject	IRISRelationshipObject
CacheMethodSignature	IRISMethodSignature
CacheStream	IRISStream
CacheFileCharacterStream	IRISFileCharacterStream
CacheFileBinaryStream	IRISFileBinaryStream
ICacheObject	IIRISObject
CacheIStruct	IRISIStruct
ICacheCollection	IIRISCollection

Assorted data types

Caché / Ensemble	InterSystems IRIS
CacheDbType	IRISDbType
CacheDate	IRISDate
CacheTime	IRISTime
CacheDateTime	IRISDateTime
CacheSysListReader	IRISSysListreader
CacheSysList	IRISSysList
CacheStatus	IRISStatus

Arrays

Caché / Ensemble	InterSystems IRIS
CacheArrayOfStrings	IRISArrayOfStrings
CacheArrayOfObjects	IRISArrayOfObjects
CacheArrayOfIntegers	IRISArrayOfIntegers
CacheArrayOfDoubles	IRISArrayOfDoubles
CacheArrayOfBinaries	IRISArrayOfBinaries
CacheArrayOfStrings	IRISArrayOfStrings
CacheArrayOfDataTypes	IRISArrayOfDataTypes
CacheArrayOfStatuses	IRISArrayOfStatuses
CacheArrayOfTimes	IRISArrayOfTimes
CacheArrayOfDates	IRISArrayOfDates
CacheArrayOfMVDates	IRISArrayOfMVDates
CacheArrayOfMVDateTimes	IRISArrayOfDateTimes
CacheArrayOfBooleans	IRISArrayOfBooleans
CacheArrayOfCurrency	IRISArrayOfCurrency
CacheArrayOfDecimals	IRISArrayOfDecimals
CacheArrayOfSysLists	IRISArrayOfSysLists
CacheArrayOfObjects	IRISArrayOfObjects
CacheArrayImpl	IRISArrayImpl
CacheArrayOfDT	IRISArrayOfDT

Lists

Caché / Ensemble	InterSystems IRIS
CacheListOfIntegers	IRISListOfIntegers
CacheListOfDoubles	IRISListOfDoubles
CacheListOfBinaries	IRISListOfBinaries
CacheListOfStrings	IRISListOfStrings
CacheListOfDataTypes	IRISListOfDataTypes
CacheListOfStatuses	IRISListOfStatuses
CacheListOfTimes	IRISListOfTimes
CacheListOfDates	IRISListOfDates
CacheListOfMVDates	IRISListOfMVDates
CacheListOfDateTimes	IRISListOfDateTimes
CacheListOfBooleans	IRISListOfBooleans
CacheListOfCurrency	IRISListOfCurrency
CacheListOfDecimals	IRISListOfDecimals
CacheListOfSysLists	IRISListOfSysLists
CacheListOfObjects	IRISListOfObjects
CacheListImpl	IRISListImpl
CacheListOfDTImpl	IRISListOfDTImpl

Library collections

Caché / Ensemble	InterSystems IRIS
CacheLibListOfObjects	IRISLibListOfObjects
CacheLibArrayOfObjects	IRISLibArrayOfObjects

Arguments

Caché / Ensemble	InterSystems IRIS
CacheObjArgument	IRISObjArgument
CacheIntArgument	IRISIntArgument
CacheDoubleArgument	IRISDoubleArgument
CacheBinaryArgument	IRISBinaryArgument
CacheStringArgument	IRISStringArgument
CacheStatusArgument	IRISStatusArgument
CacheTimeArgument	IRISTimeArgument
CacheDateArgument	IRISDateArgument
CacheTimestampArgument	IRISTimestampArgument
CacheBooleanArgument	IRISBooleanArgument
CacheDecimalArgument	IRISDecimalArgument
CacheSysListArgument	IRISSysListArgument
CacheCurrencyArgument	IRISCurrencyArgument

Return values

Caché / Ensemble	InterSystems IRIS
CacheStatusReturnValue	IRISStatusReturnValue
CacheObjReturnValue	IRISObjReturnValue
CacheIntReturnValue	IRISIntReturnValue
CacheBinaryReturnValue	IRISBinaryReturnValue
CacheBooleanReturnValue	IRISBooleanReturnValue
CacheStringReturnValue	IRISStringReturnValue
CacheTimeReturnValue	IRISTimeReturnValue
CacheDoubleReturnValue	IRISDoubleReturnValue
CacheDateReturnValue	IRISDateReturnValue
CacheTimestampReturnValue	IRISTimestampReturnValue
CacheCurrencyReturnValue	IRISCurrencyReturnValue
CacheDecimalReturnValue	IRISDecimalReturnValue
CacheSysListReturnValue	IRISSysListReturnValue

Exceptions

Caché / Ensemble	InterSystems IRIS
CacheObjException	IRISObjException
CacheStatusException	IRISStatusException
CacheMetaInfoException	IRISMetaInfoException
CacheDisconnectedObjectException	IRISDisconnectedObjectException
CacheUserMethodException	IRISUserMethodException
CacheUserErrorException	IRISUserErrorException
CacheInternalException	IRISInternalException
CacheInvalidCollectionElementtException	IRISInvalidCollectionElementtException
CacheInvalidCollectionKeyException	IRISInvalidCollectionKeyException
CacheInvalidProxyException	IRISInvalidProxyException

e. Development Platforms

InterSystems Atelier version 1.2 introduces support for connecting to InterSystems IRIS (all versions) and continues to support connecting to InterSystems Caché and Ensemble instances.

InterSystems Studio continues to support connecting to instances of the version of the product it was shipped with and to earlier versions of the same product (either Caché and Ensemble or InterSystems IRIS).

This support policy can be summarized as follows:

	InterSystems IRIS (version n)	InterSystems Caché or Ensemble (version n)
InterSystems Atelier < 1.2	Not supported	~~Supported for n ≥ 2016.2~~
InterSystems Atelier ≥ 1.2	Not supported	~~Supported for n ≥ 2016.2~~
InterSystems IRIS Studio (ship with version m)	Supported when m ≥ n	Supported when m ≥ n
InterSystems Studio (shipped with C/E)	Not supported	Supported for all versions

Summary: Key Differences and Recommendations

The following table summarizes the key differences that are likely to affect a migration of an existing Caché or Ensemble application to InterSystems IRIS. See the corresponding sections above for more details.

Difference	Recommendation
Platforms & Browsers	If your application requires support for some older operating systems and browsers, please contact the WRC at InterSystems for assistance.
File and Parameter Names	If your application uses scripts developed for installation, operations, or configuration, you should scan these for areas where you need to change names or parameters.
Class and API Names	Plan to update your application code with new names to migrate it to InterSystems IRIS.
Language Bindings	If your application uses bindings for Java, Python, Perl, C++ or Light C, plan to update it to the suggested alternative. For the .Net Bindings a separate download is available from the WRC download server.
Globals API, Caché DB Library	If your application uses any of these, plan to update it to the suggested alternative.
Caché Direct	Existing applications should move away from using the technology. Existing customers may continue to use Caché Direct, but will require a special license key.
Zen Web Development	Your Zen applications will work on InterSystems IRIS, but you should consider porting or re-implementing them now or in the near future.
Zen Reports, DeepSee Visual Reporting	Applications using these will work on InterSystems IRIS, but you should work with InterSystems on a plan to migrate away from these.
TCP Provider	This is not provided in InterSystems IRIS; if your application uses it, you should plan to update this element of your application to another TCP provider.
Shadowing	This is not provided in InterSystems IRIS; if your application uses it, you should plan to update your application to use mirroring instead.
ASTM, DICOM, HL7, and ITK	Use InterSystems IRIS for Health.

License Keys – Special Considerations

In the section Legacy Facilities and Modules, a small number of technologies, while deprecated for new projects, might be available for existing projects that convert from Caché or Ensemble to InterSystems IRIS. To enable those technologies certain features must be enabled in the license key.

If you determine that your project will require such a technology, please let InterSystems Order Processing or your Account representative know.

Caché or Ensemble Features that require a special license key with InterSystems IRIS.

Technology	Deprecated Technology	Security Risk	Comment
Caché Direct (VisM)	yes	yes	
WebLink	yes	yes	
.Net Binding	yes	no	

Version History

Version	Date	Change
2.11	December 4, 2018	Initial publication
2.20	January 29, 2019	Adding compatibility support for C/E .Net Binding, ~~Deprecate Caché Basic~~, in-place conversion
2.21	February 19, 2019	Correction of DLL reference for C/E .Net Binding, SQL Delimited Identifiers
2.22	March 1, 2019	Caché Basic will not be deprecated and is support with InterSystems IRIS
2.25	April 22, 2019	Non-Unicode (8-bit) database format will be supported. Migration Sequencing for migrations and in-place conversions. Special license key requirements for Caché Direct and WebLink.
2.26	May 1, 2019	Coexistence & Compatibility Section: Added details for technology that will be compatible between InterSystems IRIS and C/E; updated table with minimum required C/E versions. Indicate that VISM and WebLink need special license keys
2.27	May 31, 2019	Minor clean up. Mark technologies that will require special license key features to enable them. Added section describing how to order license keys.
2.28	June 26, 2019	Updates for in-place conversion sequencing
2.30	July 31, 2019	Co-existence: ECP now supported between the two product lines. C/E Online Backups are not compatible with IRIS. ENSLIB is mounted read-only on InterSystems IRIS. General improvements for clarification.
2.31	August 1, 2019	Adjust contents for the upcoming GA of 2019.1.1.
2.40	October 11, 2019	General updates and minor adjustments
2.41	October 21, 2019	Updated Co-existence: ECP clarification, Web Gateway and InterSystems IRIS Studio full compatible between C/E and InterSystems. LDAP compatible as of C/E 2018.1.3 and 2019.1.1 (not yet released)
2.42	December 11, 2019	Updated IRIS Migration Initiative – Sequencing; %Service_WebLink reintroduced with 2019.1
2.43	January 30, 2020	SQL operator precedence. Clarification on moving Failover Mirrors from CE to InterSystems IRIS. Clarification that development platforms are not supported with the in-place migration.
2.44	Mary 5, 2020	Removed version details for features that require special licenses to enable them. In-place migration to InterSystems IRIS for Health are available Clarification for command line arguments using /bin/iris on Windows
2.45	July 17, 2020	Clarification on irisusr with respect to the in-place migration
2.46	September 30, 2020	DirectX instruction on where to get the required libraries. Introduction of a new table to specify which technologies have special license requirements.

2.47	March 8, 2021	Clarification that System settings are not export/import compatible across product lines. Updated conversion availability table. Deprecation of Atelier.
2.48	April 15, 2021	Minimum required Caché/Ensemble version to convert to IRIS is 2017.1; Clarification on ADO
2.49	September 23, 2021	Naming convention: replace adoption with migration
2.82	October 28, 2021	(Version jump intentional) Clarification in the License Key Special Considerations sections, adding Object Script $ETRAP to the deprecated list for new applications.
2.83	December 7, 2021	Clarification the CSP application should continue to use language=*Cache* and not use *objectscript*.
2.84	February 25, 2022	Fixed broken hyperlinks
2.85	March 4, 2022	Add .Net Binding to: License Keys – Special Considerations
2.86	April 13, 2022	Adjusting language for Deprecated to Discontinued where appropriate. In this document update: Shadowing
2.87	June 8, 2022	Update document to include InterSystems IRIS 2022.1
2.88	December 28, 2022	Fixed broken hyperlinks. Added additional information in the Different Names section to provide guidelines on how DLLs and other files in the installation directory are renamed in InterSystems IRIS, plus new names for the ^Buttons and ^pButtons system monitoring and diagnostic tools. Minor clarifications.