Server Migration Guide

Version 2023.3
2024-05-16

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

Server Migration Guide

Many possible scenarios exist where you want to take an existing instance of InterSystems IRIS® data platform and move it from one server to another. For example, if you want to move to a new server that is a similar or identical to the original server, you can use a full system restore.

Other scenarios can be more challenging, such as when the existing server is running an operating system that is no longer supported and you need to move the instance to a new server running a different operating system. This document provides some guidelines that will help you migrate your instance and its data to a new server. Moving data or code contained in an InterSystems IRIS database is straightforward, but you must also consider configuration settings, security settings, Task Manager tasks, and other items that need to be moved over manually.

This document is intended to be used for migrating an instance of InterSystems IRIS to a new server running the same version of InterSystems IRIS. For other types of migration, see Notes on Migrating to a Newer Version of InterSystems IRIS.

Important: As a system administrator you should review this document, decide which items are applicable to your system and whether items not on the list must also be migrated.

All InterSystems IRIS installations are unique. This document covers the topics InterSystems believes to be the most common, however, it cannot cover all possible scenarios. If you need assistance in determining the best course of action to take, contact the InterSystems Worldwide Response Center.

InterSystems recommends that customers have a document that outlines how to completely rebuild their instances. If you do not have such a document, a migration event is a good time to create such a document.

Important: If you are migrating to an operating system that does not support your current version of InterSystems IRIS, you should first perform an upgrade to a supported version of InterSystems IRIS. (See Upgrading InterSystems IRIS.) After fully testing this upgrade, perform the migration to the new operating system.

Note: This document is intended to be used for InterSystems IRIS migrations only. While the general principles apply to other InterSystems products, those products may require unique procedures that are beyond the scope of this document.

This document:

- Describes the high-level steps needed to set up a new server prior to performing the migration.

- Alerts you to some special considerations pertaining to migration to a different operating system.

- Provides guidelines on migrating the types of items listed below:

  - InterSystems IRIS license key

  - Configuration Parameter File

  - Databases

  - Namespace interoperability settings

  - Security settings (including users, roles, resources, services, and applications)

  - Task Manager tasks

  - Custom code in %SYS namespace

  - CSP, JS, and CSS files

–   External linked libraries and custom shared libraries

- Describes how to migrate mirror members.

- Provides some notes on migrating to a later version of InterSystems IRIS.

1 Before Performing the Migration

1.1 Provision the Target Server

Unless you are purposely downsizing your server, make sure that the target server equals or exceeds the source server in terms of:

- CPU

- Memory

- Disk space

For more information on sizing a server, see System Resource Planning and Management.

1.2 Perform the Initial Installation

After procuring the target server, perform the initial installation of the operating system and InterSystems IRIS on the target server:

1. Install the operating system.

2. Perform the following:

    - Make sure necessary users at the operating system-level are configured.

    - Make sure large or huge pages are configured.

    - Make sure swap space or page file configurations are consistent with those on the source server.

    - Make sure all drives or filesystems are available with the same names and permissions.

    - Make sure the new machine has the same IP address, or if it has a new IP address then consider if other systems need to allow the new IP address through their firewalls.

3. Install the same version of InterSystems IRIS as on the source server.

    For the smoothest migration, make sure that the following installation options are the same as on the source server:

    - Instance name

    - Installation directory

        Note: If you do change the installation directory, note that you may have to update any directory path names in the CPF and the exported security settings (such as the Applications settings and the SSL Configurations settings) when you perform the migration.

    - Setup type (for example, Development, Server, or Custom)

    - Character width (8-bit or Unicode)

- Port numbers

- Security Setting (Minimal, Normal, or Locked Down)

    For more information, see Installation Planning Considerations.

4. Install the web server and configure the web server and the web gateway (if required). For more information, see the Web Gateway Guide.

5. Install any necessary security certificates.

6. Configure any needed firewall settings.

2 Notes on Migrating from One Operating System to Another

If you are migrating from one operating system to another, for example, from Windows to Linux, some files and settings may not be directly compatible. For example, you may need to hand-edit your configuration parameter file (CPF) or exported settings files before migrating them to the target server.

Before you begin the migration process, make a backup copy of the original configuration parameter file, /<installdir>/iris.cpf, on the target server, in case you need to refer to it later. Also, export the default security settings on the target server to an XML file, using the **Security.System.ExportAll()** method. (See Use the Security.System.ExportAll() Method later in the document.)

Items you may need to edit include, but are not limited to the following:

- Any directory path names in the CPF, the exported Applications settings, and the exported SSL Configurations settings

- The overview parameter in the [Config] section of the CPF

See the original CPF and exported settings files on the target server for guidance.

If you are migrating to an operating system that has a different Endianness (big-endian versus little-endian), you must first convert the byte order of any databases using the cvendian utility. See Using cvendian for Byte Order Conversion for details.

Important: If you did not purchase a platform-independent license, contact your account team to obtain a license for the new platform before performing the migration.

3 Perform the Migration

3.1 InterSystems IRIS license key

When you are ready to begin the migration, perform a clean shutdown of the instances on both servers. Do not restart the instances until instructed.

Copy the iris.key file from the /<installdir>/mgr/ directory on the source server to the same directory on the target server.

3.2 Configuration Parameter File (CPF)

Copy the iris.cpf file from the /<installdir>/ directory on the source server to the same directory on the target server.

Important: If the target server is running on a different operating system from the source server, see Notes on Migrating from One Operating System to Another.

3.3 Databases

3.3.1 Decide Which Databases to Migrate

In general, it is easy to migrate a database from the source server to the target server simply by copying it, but not all databases can or should be migrated.

User-Created Databases

Any user-created databases that you want to retain on the target server should be migrated.

IRISSYS

The IRISSYS database, located in the /<installdir>/mgr directory, cannot be migrated by copying it. Any data needed from the source IRISSYS database, such as security settings or tasks, can be exported and then imported into the destination IRISSYS database. See the remainder of this document for details.

Other System-Supplied Databases

Other system-supplied databases should not be migrated, as fresh copies are provided by the installer. These databases include:

- ENSLIB
- IRISAUDIT
- IRISLIB
- IRISLOCALDATA
- IRISTEMP

If you have placed custom code in the ENSLIB database, InterSystems recommends evaluating and implementing alternative solutions before performing the migration.

3.3.2 Migrate the Databases

Copy the IRIS.DAT files to be migrated, along with their enclosing directories to the same locations on the target server.

Note: If you see any iris.lck files with the iris.dat files on the source server, the source instance of InterSystems IRIS is running or did not shutdown cleanly. Shut down the instance cleanly and try again.

Start the instance on the target server, and it will start up using the new license key, CPF, and databases.

Note: If InterSystems IRIS cannot mount a database upon startup, you will see the following message:

```
Database <db_name> is required, but could not be mounted
Shutting down the system
```

Frequently, this error means one of the following:

- You did not copy the database to the location you intended to copy it.

  Make sure that the database is located in the correct directory, and then start InterSystems IRIS again.

- You installed InterSystems IRIS into a different directory than on the source server.

  Edit the CPF and change all directory names to indicate the new installation directory, and then start InterSystems IRIS again. You will also likely need to edit any exported settings files (such as the Applications or SSL Configurations settings) to fix the directory names before importing them into the target instance of InterSystems IRIS. See Security Settings.

- The file or directory permissions are not set correctly for the InterSystems IRIS owner and users.

  Adjust the permissions and then start InterSystems IRIS again.

Note: If you see a message similar to the following upon startup, it means an iris.lck file was present when you copied over the database:

```
Database <db_name> is locked from <instance_name>
Cannot remove database lock <file_name>
```

The iris.lck file prevents multiple instances from mounting the same database with read/write permissions at the same time. If you accidentally copied the iris.lck file from the source server and you are certain no other instance has mounted the database already, you can delete it.

3.4 Namespace Interoperability Settings

If you are using interoperability productions in a namespace, you must enable the namespace to use productions on the target server, as this setting does not come over in either the CPF or the databases associated with the namespace.

Run the following command for each namespace to be interoperability enabled:

```
do ##class(%Library.EnsembleMgr).EnableNamespace("<namespace>",1)
```

As an alternative, you could reconfigure the namespaces in a single operation using the configuration merge feature. To do this, execute the **iris merge** command with a merge file like the following, containing an entry for each namespace you want to be interoperability enabled:

```
[Actions]
ModifyNamespace:Name=NamespaceA,Interop=1
ModifyNamespace:Name=NamespaceB,Interop=1
ModifyNamespace:Name=NamespaceC,Interop=1
. . .
```

3.5 Security Settings

Migrate your security settings using one of the following methods:

- Security.System.ExportAll()

- Individual Security Package Export Methods

- ^SECURITY Routine

Start InterSystems IRIS on both the source and target servers before beginning this step.

Important: If the target server is running on a different operating system from the source server, see Notes on Migrating from One Operating System to Another.

Important: If you are migrating SSL configurations, make sure that you copy your certificate files into the same locations on the target server before importing the configurations. If the certificates are not in place, the import will fail.

3.5.1 Use the Security.System.ExportAll() Method

You can run the method **Security.System.ExportAll()** to export the following types of security records from the source server:

- Applications
- Documentation databases
- Events
- KMIP servers
- LDAP configuations
- Open AM identity services
- Phone providers
- Resources
- Roles
- Services
- SQL privileges
- SSL configurations
- System
- Users
- X509 credentials
- X509 users

In the %SYS namespace, run the following command:

```
set status = ##class(Security.System).ExportAll()
```

Note: By default, the **ExportAll()** method writes the settings to the file /<installdir>/mgr/SecurityExport.xml.

You can use the Flags argument to export a subset of these settings.

See **Security.System.ExportAll()** in the class reference documentation for details.

Copy the resulting XML file to the target server and run the method **Security.System.ImportAll()** to import the settings.

In the %SYS namespace, run the following command:

```
set status = ##class(Security.System).ImportAll()
```

Note: By default, the **ImportAll()** method reads the settings from the file /<installdir>/mgr/SecurityExport.xml.

You can use the `Type` argument to import a subset of the exported settings.

See **Security.System.ImportAll()** in the class reference documentation for details.

3.5.2 Use the Individual Security Package Export Methods

You can use the individual export methods of the Security package to selectively choose which settings to migrate.

See Tips on exporting and importing security settings for a list of applicable classes.

See the class reference documentation for the Security package for details on the individual classes.

This sample method can be run on the source server to export the specified settings to a series of XML files in the C:\TEMP directory.

```
ClassMethod ExportSecuritySettings(directory as %String = "C:\TEMP\Users")
{
    new $namespace
    set $namespace = "%SYS"
    do ##class(Security.Resources).Export(directory _ "\Resources.xml")
    do ##class(Security.Services).Export(directory _ "\Services.xml")
    do ##class(Security.Roles).Export(directory _ "\Roles.xml")
    do ##class(Security.Users).Export(directory _ "\Users.xml")
    do ##class(Security.Applications).Export(directory _ "\Applications.xml")
    /* additional exports */
}
```

Note: If no location is specified, the **Export()** methods in the individual classes in the Security package export the settings to XML files in the directory /<installdir>/mgr.

Copy the exported XML files to the same location on the target server and run the following sample method on the target server to import the settings.

```
ClassMethod ImportSecuritySettings(directory as %String = "C:\TEMP\Users")
{
    new $namespace
    set $namespace = "%SYS"
    do ##class(Security.Resources).Import(directory _ "\Resources.xml")
    do ##class(Security.Services).Import(directory _ "\Services.xml")
    do ##class(Security.Roles).Import(directory _ "\Roles.xml")
    do ##class(Security.Users).Import(directory _ "\Users.xml")
    do ##class(Security.Applications).Import(directory _ "\Applications.xml")
    /* additional imports */
}
```

Important: When you import your security settings, order matters. For example, if you try to import a role that uses a particular resource and the resource has not yet been imported, an error will occur. Import resources before roles, and roles before users.

Note: If no location is specified, the **Import()** methods in the individual classes of the Security package import the settings from XML files in the directory /<installdir>/mgr.

3.5.3 Use the ^SECURITY Routine

You can also run the **^SECURITY** routine from the %SYS namespace to interactively export security settings to XML files in much the same way as using the methods in the **Security** package.

To migrate the security settings, copy the XML files to the target server and run the **^SECURITY** routine from the %SYS namespace to interactively import the security settings from the XML files

Important: When you import your security settings, order matters. For example, if you try to import a role that uses a particular resource and the resource has not yet been imported, an error will occur. Import resources before roles, and roles before users.

Note: By default, the **^SECURITY** routine exports the settings to XML files in the directory /<installdir>/mgr and imports them from the same directory.

Note: To ensure that migration is a repeatable process, InterSystems recommends using the API calls described in the sections Security.System.ExportAll() or Individual Security Package Export Methods, above. The **^SECURITY** routine is more suitable for ad hoc use.

See Tips on exporting and importing security settings for more information on using the **^SECURITY** routine.

3.6 Task Manager Tasks

Export any Task Manager tasks you want to migrate by exporting individual tasks from within the Task Manager or by passing a list of tasks to the **%SYS.Task.ExportTasks()** method.

To use the Task Manager to export a task:

1. In the Management Portal, select **System Operations** > **Task Manager** > **Task Schedule**.

2. On the Task Schedule page, click the name of the task you want to export.

3. On the Task Details screen, click **Export**.

4. On the Export screen, enter a path and file name, and click **Perform Action Now**.

Note: By default, Task Manager exports a task to the file /<installdir>/mgr/Temp/ExportTask.xml.

You can also use the Task Schedule page to identify the IDs of the tasks you want to export. Then set a variable to the list of task IDs and call **%SYS.Task.ExportTasks()** to export the desired tasks to an XML file.

For example, from any namespace:

```
set tasklist = $lb(1, 1000, 1001)
set status = ##class(%SYS.Task).ExportTasks(tasklist, "Tasks.xml")
```

Note: **%SYS.Task.ExportTasks()** exports tasks to an XML file with the specified name in the directory of the Default Database for Globals for the namespace where you ran the command.

Copy the exported XML files to the target server and then import them using the Task Manager:

1. In the Management Portal, select **System Operations** > **Task Manager** > **Import Tasks**.

2. On the Import screen, enter a path and file name, and click **Perform Action Now**.

Note: By default, Task Manager imports tasks from the directory /<installdir>/mgr.

If you are importing a System task, Task Manager will create a new task, rather than overriding an existing task. The new task will have the next available ID greater or equal to 1000. If desired, delete or suspend the existing task from the Task Manager.

When you are done importing your tasks, you need to configure your Task Manager email settings on the target server, as these do not come over in the Task Manager export. You can do this in the Management Portal by going to **System Administration** > **Configuration** > **Additional Settings** > **Task Manager Email**. Copy and paste each field from the same screen on the source server.

3.7 Custom Items in %SYS Namespace

Export any custom classes, routines, or globals from the %SYS namespace of the source server and import them to the %SYS namespace on the target server.

For example, migrate the following routines if you are using them:

- **^ZWELCOME**

- **^ZAUTHENTICATE** and **^ZAUTHORIZE**

- **^ZMIRROR**

- **^%ZSTART** and **^%ZSTOP**

- **^%ZLANG***

- **^%ZJREAD**

3.8 CSP, JS, and CSS Files

Copy all files and directories for any custom CSP applications from the source server to the same locations on the target server.

3.9 External Linked Libraries and Custom Shared Libraries

If you are using any external linked libraries or custom shared libraries, copy these from the source server to the target server.

This procedure is site specific and instructions are not provided in this document.

4 Migrate Mirror Members

If you are using InterSystems IRIS mirroring, you can use migration to assist you in:

- Creating a New Mirror Member

- Moving to a Mirror to New Servers

For an in-depth discussion of mirroring, see Mirroring.

4.1 Use Migration to Assist in Creating a New Mirror Member

If you have an existing mirror, you can use migration to set up a new server to be used as a new failover member or async member. Simply migrate an existing failover member to a new server, and then join the target server to the mirror.

Prepare the target server as described in this document, with the following differences.

- Delete the `[Mirrors]` and `[MapMirrors]` sections from the CPF before migrating it to the new server.

- Revert the `[MirrorMember]` section of the CPF to its default before migrating it to the new server. You can copy this section from the freshly installed CPF on the target server.

Note: This prevents the new instance from trying to use the same Agent Address as the old instance.

Then on the new InterSystems IRIS instance, perform the following:

1. Start the ISCAgent. (See Starting and Stopping the ISCAgent.)

2. In the Management Portal, select **System Administration** > **Configuration** > **Mirror Settings** > **Enable Mirror Service**.

3. On the Edit Service window, select **Service Enabled**, and then click **Save**.

4. Under Mirror Settings, select **Join as Failover** or **Join as Async**, depending on which type of mirror member you would like to add.

   Note: If you don't see the **Join as Failover** or **Join as Async** menu options, but instead see the **Edit Mirror** option, it indicates that some mirror configuration information remained in the CPF before it was migrated. You will need to remove the existing mirror configuration, as follows:

   a. Select **Edit Mirror**.

   b. Click **Remove Mirror Configuration**.

   c. On the Remove Mirror Configuration screen, click **Remove**. Do not remove the mirrored attribute from the mirrored databases.

   d. Restart InterSystems IRIS and resume with Step 4.

5. On the Join Mirror screen, fill in the appropriate information for the existing mirror, and then click **Next**.

6. After the system retrieves information from the mirror, confirm the Member Information is correct.

7. If you are adding an async member, select the desired **Async Member System Type**.

8. Click **Save**.

9. Select **System Operation** > **Mirror Monitor**.

10. On the Mirror Monitor screen, under Mirrorred Databases, **Activate** each mirrored database.

11. Then, **Catchup** each mirrored database.

4.2 Use Migration to Assist in Moving to a Mirror to New Servers

Migration can also assist you in moving an existing mirror to new servers.

For example, you might have a mirrored configuration, on operating system X, where Server A is the primary failover member and Server B is the backup failover member. Your goal is to migrate to a configuration on operating system Y, where Server C is the primary failover member and Server D is the backup failover member.

The procedure for performing this move is summarized in the table below:

Table 1: Moving a Mirror to New Servers

Step	Server A	Server B	Server C	Server D
Perform migration to Server C and Server D, and add as DR async members (See Use Migration to Assist in Creating a New Mirror Member).	Primary	Backup	DR Async	DR Async
Promote C to failover member, making C the backup and demoting B to DR async (See Promoting a DR Async Member to Failover Member)	Primary	DR Async	Backup	DR Async
Failover from A to C, making A the backup and C the primary (See Planned Failover to a Promoted DR Async)	Backup	DR Async	Primary	DR Async
Promote D to failover member, making D the backup and demoting A to DR async (See Promoting a DR Async Member to Failover Member)	DR Async	DR Async	Primary	Backup
Remove A and B from mirror (See Editing or Removing an Async Member)	Disconnected	Disconnected	Primary	Backup

5 Notes on Migrating to a Newer Version of InterSystems IRIS

There are times when you may want to perform a migration to a server running a newer version of InterSystems IRIS, for example, when evaluating how your application runs on the newer version of InterSystems IRIS. Some additional steps may be required for this type of migration, however, the details are beyond the scope of this document. If possible, perform a standard upgrade, instead of migrating to a newer version of InterSystems IRIS.

For information on migrating from Caché or Ensemble to InterSystems IRIS, see *How to Migrate to InterSystems IRIS*, available on the WRC Document Distributions page (login required).