InterSystems™
IRIS Data Platform

System Administration Guide

Version 2023.1
2024-07-11

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

System Administration Guide
InterSystems IRIS Data Platform Version 2023.1 2024-07-11

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

List of Tables

1

Using the Management Portal

InterSystems IRIS® data platform enables you to perform system administration and management tasks via a web application, the InterSystems Management Portal.

1.1 Starting the Management Portal

You can start the Management Portal in the following ways:

- On Microsoft Windows platforms, click **Management Portal** on the InterSystems IRIS launcher menu.
- In your web browser go directly to the Management Portal URL, in the following form:

  ```
  http://machine:port/csp/sys/UtilHome.csp
  ```

  Where *machine* is the IP address of your system (such as `localhost`) and *port* is the port number of the web server configured for use by InterSystems IRIS. For example, the default location on the local server of a single installation of InterSystems IRIS is *http://localhost:52773/csp/sys/UtilHome.csp* (where `52773` is the default InterSystems IRIS web service server port number).

For information about the Management Portal, see Management Portal Overview.

Depending on the authentication settings for your system, you may have to log in before going to the Portal home page. See Management Portal Login Page for an overview of login requirements. In addition, each functional area requires access to particular resources; see Management Portal Web Application Structure.

Important: *Microsoft Windows 2003 Users Trusted Site Security Setting* — The first time you visit the Management Portal, you may receive a warning about the web site being blocked. When prompted, add the site to the trusted sites zone. InterSystems also recommends you allow session cookies for Portal procedures to function properly.

1.1.1 Management Portal Login Page

Whether or not you must enter a username and password to use the Portal depends on the authentication settings of the Management Portal web application (/csp/sys). There are two conditions:

Unauthenticated Access Only

Neither requires nor accepts a username and password — If the Portal accepts only unauthenticated connections (the default setting for Minimal security installs), you do not require a username and password to access the Portal; you bypass the **Login** page when you use the methods in the previous section to start the Portal. If navigation does bring you to the login page (by clicking **Logout**, for example), you see the following message:

Please click here to log in.

Authenticated Access

Requires a username and password — If your security settings require authentication for the Management Portal web application and you are not already authenticated on the system, the login page displays asking you to enter a **User Name** and **Password**. After entering these fields, click **Login** to display the Management Portal home page.

Note: To change your password, click your name in the Management Portal Header, which displays the change password dialog box.

Important: You can look up the **Authentication allowed** settings on the **Web Applications** page (**System Administration** > **Security** > **Applications** > **Web Applications**) by clicking **Edit** in the /csp/sys application row.

1.1.2 Management Portal Web Application Structure

The main Portal web application is /csp/sys. To provide enforcement of privileges within the Management Portal, the Portal is further split into distinct web applications. The Management Portal itself is not responsible for preventing non-privileged users from performing actions; this is handled by the system API methods that the Portal calls. The Portal does, however, attempt to keep non-privileged users out of restricted pages to prevent `<PROTECT>` errors.

There are four types of user for the Portal, which roughly correspond to predefined resources within InterSystems IRIS:

User Type	Resource	Web Application	Tasks
Security Manager	%Admin_Secure	/csp/sys/sec	View and edit list of users, roles, and other security tasks.
Manager	%Admin_Manage	/csp/sys/mgr	Change system configuration and define backup sets.
Operator	%Admin_Operate	/csp/sys/op	View system status pages and perform backups.
Explorer (public user)	%Development	/csp/sys/exp	View home page, view classes, routines, and globals, and use SQL pages, provided the user has access to the appropriate resources.

1.2 Management Portal Overview

This section describes some common layout elements of Management Portal pages. The following topics are discussed:

- Management Portal Home Page
- Management Portal Header
- Management Portal Ribbon

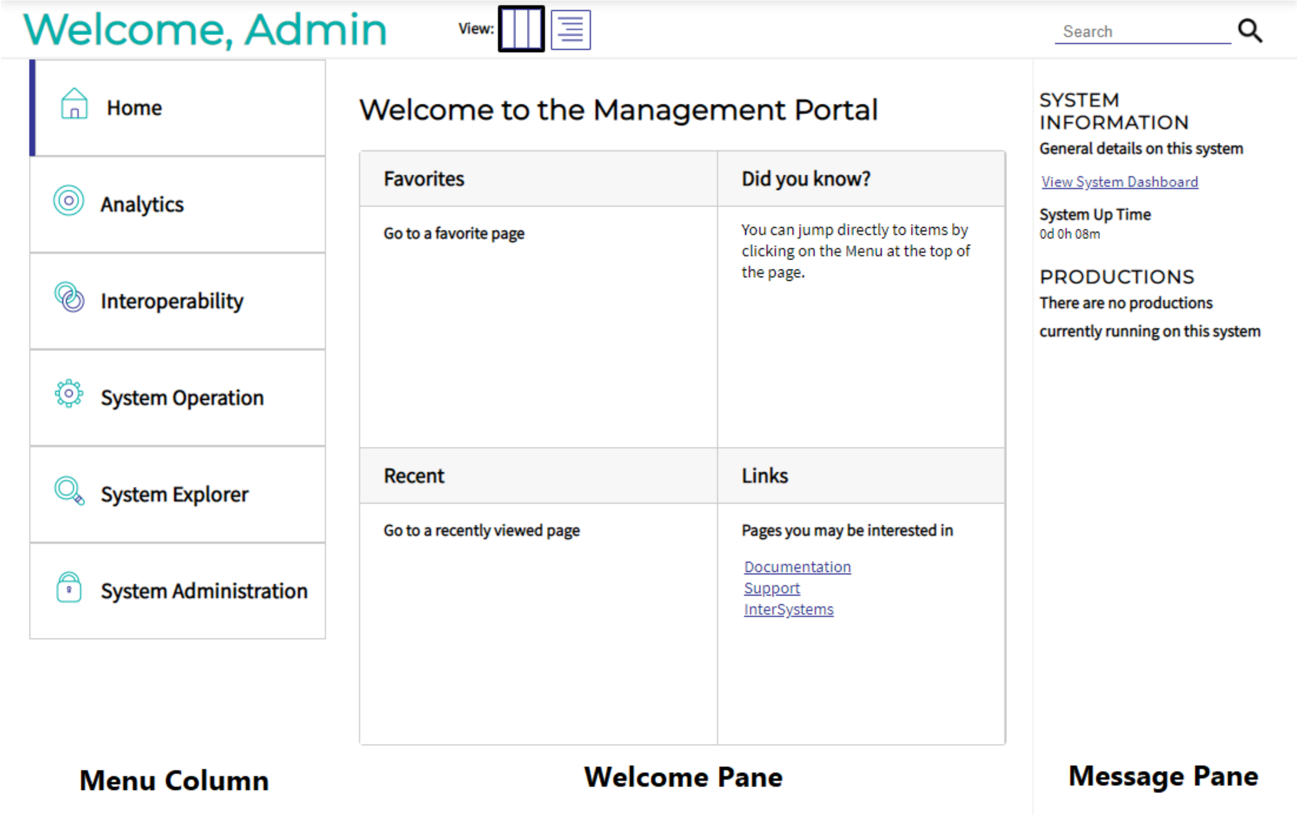
- System Overview Information

Note: Anywhere in the Management Portal, moving your cursor over a menu item displays a description of that item.

1.2.1 Management Portal Home Page

The Management Portal home page is titled **Welcome, <user>**. Next to the title, the ribbon contains the following options:

- Two **View** buttons, which let you specify how to display links in the menu column.
- A search bar, located on the right side of the ribbon. When you specify a word and press **Enter**, a list of all pages containing that word is displayed; then, you can click the destination page you want to display without having to navigate through the submenus.



Welcome, Admin

Welcome to the Management Portal

Favorites

Go to a favorite page

Did you know?

You can jump directly to items by clicking on the Menu at the top of the page.

Recent

Go to a recently viewed page

Links

Pages you may be interested in

Documentation
Support
InterSystems

SYSTEM INFORMATION

General details on this system

View System Dashboard

System Up Time
0d 0h 08m

PRODUCTIONS

There are no productions currently running on this system

The following sections describe the areas of the home page:

- Management Portal Menu Column
- Management Portal Welcome Pane
- Management Portal Message Pane

1.2.1.1 Management Portal Menu Column

The menu column, located on the left edge of the home page, is the primary method of navigating the Portal.

For detailed information about navigating the Management Portal menus, see Navigating the Management Portal.

For information about specific pages of the Management Portal, see Management Portal Page Reference.

1.2.1.2 Management Portal Welcome Pane

The welcome pane occupies the center of the home page, and includes shortcuts to pages you visit frequently. It contains the following fields:

- **Favorites** — Lists Management Portal pages you have selected as favorites (see Action Pane); you can click each page title to go directly to that page.
- **Recent** — Lists the most recent pages you have displayed since the last time InterSystems IRIS was started.
- **Did you know?** — Displays tips.
- **Links** — Links to pages you might want to visit.

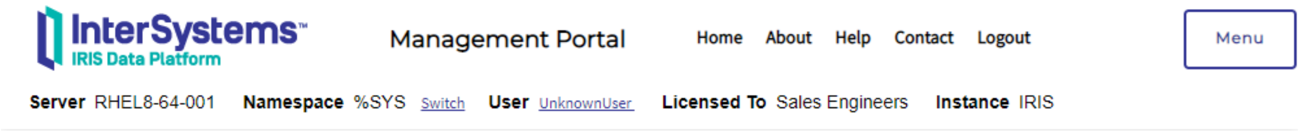
1.2.1.3 Management Portal Message Pane

The message pane, located on the right edge of the home page, displays general system information and provides a link to the system dashboard. For more information about the dashboard, see Monitoring InterSystems IRIS Using the Management Portal.

If the instance is a mirror member, the message pane also displays the mirror it belongs to, its status and member type, and a link to the Mirror Monitor (see Monitoring Mirrors).

1.2.2 Management Portal Header

The header, located at the top of every page in the Management Portal, can be used to navigate the Portal quickly.



Server RHEL8-64-001 **Namespace** %SYS **User** UnknownUser **Licensed To** Sales Engineers **Instance** IRIS

The header contains the following links:

- **Home** — Displays the Management Portal home page.
- **About** — Displays system overview information.
- **Help** — Displays the online documentation (help) for the page/topic you are viewing.
- **Contact** — Displays a contact page for the InterSystems Worldwide Response Center (WRC).
- **Logout** — Logs you out and brings you to the **Login** page of the Management Portal.
- **Menu** — Displays a list of common tasks based on the roles the user holds.

The header also contains useful information, such as:

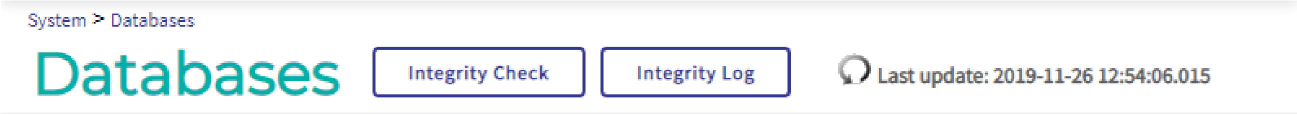
- **Server** — The name of the server running InterSystems IRIS.
- **Namespace** — The name of the namespace currently being used. To work in a different namespace, click **Switch** and select the desired namespace.
- **User** — The name of the user logged in to the Management Portal. To change the password for the user, click the name.
- **Licensed to** — The customer name that appears in the license key information.
- **Instance** — The name of the InterSystems IRIS instance running on the server.

In addition, you can display a **System Mode** label (for example, **Test System**); for information, see Memory and Startup Settings.

The left side of the Management Portal header displays the name of the product you are using.

1.2.3 Management Portal Ribbon

The ribbon lies directly beneath the header, and displays different contents specific to each page. As an example, the ribbon from the **Databases** page (**System Explorer** > **Databases**) is pictured below:



System > Databases

Databases

Last update: 2019-11-26 12:54:06.015

Typical contents of the ribbon are:

- The title of the Management Portal page being displayed.
- The breadcrumbs to the current page, listed directly above the page title. Each page listed in the path is an active link, which you can use to return to a previously displayed submenu/list. When you have made unsaved changes on a page, an asterisk is appended to the breadcrumbs, such as **System > Configuration > Memory and Startup — (configuration settings)***. You are always prompted for confirmation before navigating away from unsaved changes.

  Note: The breadcrumbs do not list every page in the path, and the pages in the breadcrumbs do not always match those in the navigation menu. You can always navigate to a specific page by clicking **Home** to return to the Management Portal home page and using the search tool, which is described later on in this section.

- Several buttons that allow you to perform operations on the page. For example, the **Databases** page contains the buttons **Integrity Check** and **Integrity Log**.
- A refresh button, with information about when the page was last updated.

1.2.4 System Overview Information

When you click **About** on the header of the Management Portal, a table displays with the following information:

- **Version** — Specific build information for this instance of InterSystems IRIS including platform, build number, and build date.
- **Configuration** — Name and location of the configuration (.cpf) file this instance is using.
- **Database Cache (MB)** — Space allocated for databases.
- **Routine Cache (MB)** — Space allocated for routines.
- **Journal file** — Name and location of current journal file.
- **SuperServer Port** — Port number on which the InterSystems IRIS server is running.
- **Web Server Port** — Port number on which the web server is running.
- **License Server Address/Port** — IP address of the InterSystems IRIS license server and port number on which it is running.
- **Licensed to** — Customer name that appears in the license key information.
- **Cluster support** — Indicates whether or not this instance is part of a cluster.
- **Mirroring** — Indicates whether or not this instance is a member of a mirror.
- **Time System Started** — Date and time this instance of InterSystems IRIS last started.
- **Encryption Key Identifier** — If encryption is activated, the GUID (global unique ID) of the encryption key.

- **NLS Locale** — National Language Support locale. For more information, see the Using System Classes for National Language Support and Using the NLS Pages of the Management Portal.
- **Preferred language for this session** — Drop-down list of languages in which the Management Portal has been localized and can be displayed. You can change the display language by selecting a new one from the drop-down. Initially, the preferred language for the browsing session is the one specified for the browser, or English if the browser language is not supported; after you select a preferred language in a particular browser, it is used by the Management Portal in that browser even if the browser language is changed.

1.3 Navigating the Management Portal

The Management Portal has two different view modes to choose from for page navigation. You can switch between view modes at any time by clicking the desired **View** button in the Management Portal ribbon. The view modes are:

- Columns view — Displays page in columns; this is the default mode. This view mode allows you to select favorites and assign custom resources.
- List view — Displays page names in drop-down list.

Note: In addition to using views to navigate the Management Portal, you can go directly to a page via the search tool, which is located in the ribbon; for more information, see Management Portal Ribbon.

1.3.1 Columns view

The columns view displays submenus that you use to reach a destination page. If you click on a title/name in the final submenu, the destination page is displayed; if you click inside the box but not on the title/name itself, the Action Pane is displayed on the right.

1.3.1.1 Action Pane

The action pane explains a menu option and lets you perform several actions related to it. To display the action pane, while in Columns view, click inside the box that surrounds an option in the final submenu, but not on the option itself. You can do the following in the action pane:

- Add the destination page to the **Favorites** list on the home page, as described in Management Portal Welcome Pane.
- View the system resource required to load the page and assign a custom resource, as described in Use Custom Resources with the Management Portal.

  Note: If the name of a resource is truncated in the action pane, move the pointer over the name to see it in its entirety.

- Go to the destination page by clicking **Go**.

The action pane is available only in Columns view. When you navigate to a page using list view or the search tool, you cannot add it as a favorite or assign a custom resource.

1.3.2 List View

The list view displays lists that you use to reach your destination page, but does not let you specify a destination page as a favorite, or assign a custom resource. When you select the page you want to display, click **Go**.

1.4 Management Portal Page Reference

The Management Portal is made up of a number of pages divided into the following functional areas:

- Home — Management Portal home page
- Health — Healthcare pages. Only for InterSystems IRIS for Health™ and HealthShare® Health Connect.
- Analytics — InterSystems IRIS® data platform Business Intelligence pages
- Interoperability — Productions pages
- System Operations — System operation pages
- System Explorer — Database management pages
- System Administration — System administration pages

This document provides links to the appropriate section of documentation for each page of the Portal.

1.4.1 Home

The home page contains a number of options for navigating the Management Portal. See Management Portal Home Page for more information.

1.4.2 Health

Healthcare-specific features in InterSystems IRIS for Health and Health Connect are displayed on pages available from the **Health** portion of the Management Portal. You need to install a Foundation namespace before accessing these health pages. For details, see the "Using the Installer Wizard" section of the InterSystems IRIS for Health Installation Guide or Health Connect Installation Guide.

1.4.3 Analytics

The Business Intelligence pages are divided into the categories displayed in the **Analytics** portion of the Management Portal. See Introduction to the Business Intelligence User Interfaces.

1.4.4 Interoperability

The productions pages are divided into the categories displayed in the **Interoperability** portion of the Management Portal. See Finding Information on Menu Items for more information.

1.4.5 System Operations

The system operator pages are divided into the categories displayed in the **System Operations** portion of the Management Portal. The following table displays each category and the relevant documentation source.

Menu Item	Information
System Dashboard	Monitoring InterSystems IRIS Using the Management Portal
Backup	Managing Online Backups
Databases	Maintaining Local Databases

Menu Item	Information
Processes	Controlling InterSystems IRIS Processes
SQL Activity	Monitoring SQL Activity
Locks	Monitoring Locks
Journals	Journaling
Mirror Monitor	Monitoring Mirrors
Task Manager	Using the Task Manager
LDAP Configurations	View LDAP Configurations in the Portal as %Operator.
System Logs	Monitoring Log Files
System Usage	Monitoring System Performance
License Usage	Managing InterSystems IRIS Licensing
Background Tasks	JOB
Diagnostic Reports	Using the InterSystems IRIS Diagnostic Report

1.4.6 System Explorer

The database management pages are divided into the categories displayed in the **System Explorer** portion of the Management Portal. The following table displays each category and the relevant documentation source.

Menu Item	Information
Classes	Classes
SQL	Introduction to InterSystems SQL
Routines	User-Defined Code
Globals	Formal Rules about Globals
Tools	Provides classes that you can use to record, randomize and playback HTTP-based scripts against various applications for the purpose of QA, scalability, and network load testing. See `%WebStress` in the *InterSystems Class Reference* for more information.

1.4.7 System Administration

The system administrator pages are divided into the categories displayed in the **System Administration** portion of the Management Portal. The following table displays each category and the relevant documentation source.

Menu Item	Information
Configuration	Configuration Pagestable (below)
Security	System Management and Security
Licensing	Managing InterSystems IRIS Licensing
Encryption	Encryption Guide

1.4.7.1 Configuration Pages

The Configuration Pages are a subcategory of the system administration pages.

Menu Item	Information
System Configuration	System Configuration Pages table (below)
Connectivity	Connectivity Pages table (below)
Mirror Settings	Configuring Mirroring
Database Backup	Configuring InterSystems IRIS Backup Settings
SQL and Object Settings	SQL and Object Settings Pages table (below)
Device Settings	Device Settings Pages table (below)
National Language Settings	Configuring National Language Support (NLS)
Additional Settings	Additional Settings Pages table (below)

System Configuration Pages

The System Configuration Pages are a subcategory of the configuration pages.

Menu Item	Information
Memory and Startup	Configuring System Information
Namespaces	Configuring Namespaces
Local Databases	Configuring Databases
Remote Databases	Remote Databases
Journal Settings	Configuring Journal Settings

Connectivity Pages

The Connectivity Pages are a subcategory of the configuration pages.

Menu Item	Information
ECP Settings	Deploying a Distributed Cache Cluster
SQL Gateway Settings	Creating Gateway Connections for External Sources
External Language Servers	[Gateways]

SQL and Object Settings Pages

The SQL and Object Settings pages are a subcategory of the configuration pages.

Menu Item	Information
SQL	[SQL].
Objects	Swizzling a Fieldname Property with %ObjectSelectMode=1
TSQL Compatibility	TSQL Settings
ISQL Compatibility	ISQL Configuration Settings
System DDL Mappings	System Datatypes.
User DDL Mappings	User Datatypes

Device Settings Pages

The Device Settings pages are a subcategory of the configuration pages.

Menu Item	Information
Devices	Devices *Configuration Parameter File Reference*
Device Subtypes	DeviceSubTypes
IO Settings	[IO]
Telnet Settings	[Telnet]

Additional Settings Pages

The Additional Settings pages are a subcategory of the configuration pages.

Menu Item	Information
Compatibility	[Miscellaneous]
Advanced Memory	[config] parameters
Monitor	Monitoring InterSystems IRIS Using SNMP
Source Control	*Integrating InterSystems IRIS with Source Control Systems*.
Startup	[Startup]
Task Manager Email Settings	Configuring Task Manager Email Settings in the “System Administration Guide”.

2

Configuring System Information

InterSystems IRIS® data platform stores system-wide configuration information in a file called the configuration parameter file (CPF). This file is an important tool, as it contains most of the configurable settings for an InterSystems IRIS instance. A default CPF is deployed with every instance and is located in the installation directory. You can modify the CPF directly in a text editor, or indirectly from the Management Portal or the Terminal. On UNIX® and Linux, you can also customize the CPF during deployment by specifying a configuration merge file, which InterSystems IRIS uses to update the default CPF before the instance starts for the first time. For more information about the CPF and its parameters, Introduction to the Configuration Parameter File; for information about the configuration merge feature, see Using Configuration Merge to Deploy and Reconfigure InterSystems IRIS.

There are several startup settings that you must change for any newly installed instance, and others you should review. This page discusses the settings to consider initially.

2.1 Memory and Startup Settings

When you first install InterSystems IRIS, you should review and adjust the memory allocations, among other configuration settings. There are three primary actions you must take in determining the way an InterSystems IRIS instance uses memory, as follows:

- Allocate memory to the database and routine caches, using the Memory and Startup page (**System Administration** > **Configuration** > **System Configuration** > **Memory and Startup**); see Allocating Memory to the Database and Routine Caches.
- Set the maximum memory an InterSystems IRIS process can use; see Setting the Maximum Per-Process Memory.
- Configure the size of the shared memory heap, which determines the memory available to InterSystems IRIS for purposes other than the routine and database caches; see Configuring Shared Memory Heap (gmheap).

Important: When InterSystems IRIS is first installed, database and routine cache memory allocation is set to **Automatically**, under which InterSystems IRIS allocates 25% of total physical memory for the database cache (global buffers).

This setting is not appropriate for production use. For guidelines for allocating memory to an InterSystems IRIS instance's routine and database caches and the shared memory heap, see Configuring InterSystems IRIS Shared Memory.

Other than the memory settings, the Memory and Startup page includes the following:

- **Auto-start on System Boot** — On Windows systems, an InterSystems IRIS instance is configured by default to start automatically when the host system starts. You can change this behavior, so that the instance does not start automatically on system startup, by clearing this check box.

  On UNIX ® systems, InterSystems IRIS does not provide this option and does not restart when the host system restarts. If you want InterSystems IRIS to restart when the host system restarts, use OS-level scripting as appropriate for your environment.

  Note: You may occasionally need to *prevent* IRIS from starting with the host system while the host system is down (for example during restores). This can only be done at the OS level and is dependent on your configuration.

- **Superserver Port Number** — The superserver port is the TCP port used by an InterSystems IRIS instance to accept incoming client requests. When you change this setting (which should be done with caution, as many clients may be configured to connect to the instance using this port), the change will not take effect until you restart the instance.
- **System Mode** — You can enter a label to be displayed in the Management Portal header, or select one from the drop-down list.

Click **Save** to save your modifications to these settings.

Important: Some changes on this page require an InterSystems IRIS restart and some do not. If you modify a setting that requires a restart and save your changes, none of the changes take effect until you restart InterSystems IRIS, even those that by themselves do not require a restart. If a restart is required, the message `Modification saved. You must restart system for the new values to take effect.` displays. After you close the page, the warning message does not appear again to remind you that a restart is required, and it is therefore best to restart the instance immediately.

2.1.1 Allocating Memory to the Database and Routine Caches

The database cache is system memory allocated for buffering data, and is sometimes referred to as the *global buffer pool*. The routines cache is system memory allocated for buffering routines.

When InterSystems IRIS is first installed, configuration of the database cache is set to the **Initial**, under which InterSystems IRIS allocates 25% of total system physical memory for the database cache (global buffers). *This setting is not appropriate for production use*. Before deploying the system for production use or before performing any tests or benchmarking intended to simulate production use, you must manually create an appropriate memory allocation for the database cache using the procedure described here.

By default, the configuration of the routine cache is set to **Automatic**, under which InterSystems IRIS allocates to the routine cache an amount of memory equal to 10% of the database cache for the default 8-kilobyte buffer size, with a minimum of 80 MB and a maximum of 1020 MB. For typical production instances, if the database cache size is appropriately configured, automatic allocation of the routine cache is sufficient; however, the ideal allocation for a given application depends on many factors, and adjustment may be necessary to optimize performance.

For guidelines for allocating memory on an InterSystems IRIS instance, including to the database and routine caches, see Estimating Memory Requirements.

To allocate memory to the database and routine caches:

1. On the Management Portal, navigate to the Memory and Startup page (**System Administration** > **Configuration** > **System Configuration** > **Memory and Startup**).
2. For **Configure Database Cache (Global Buffers)**, select **Specify Amount** and enter the database cache memory allocation, in megabytes.

   If you have enabled database block sizes other than the default 8-kilobyte size using the DBSizesAllowed setting on the **Startup Settings** page (**System Administration** > **Additional Settings** > **Startup**), enter an allocation for each enabled block size; the range for each size is displayed on the page. Both block size and the maximum number of buffers available have implications for performance; to determine how many global buffers InterSystems IRIS will create for a particular block size, divide the allocation for a block size by the block size. For guidelines for selecting the appropriate block sizes for your applications, see Large Block Size Considerations.

Important: If you are configuring a large ECP system, allocate at least 50 MB of 8 KB buffers for ECP control structures in addition to the 8 KB buffers required to serve your 8 KB blocks over ECP. See Increase Data Server Database Caches for ECP Control Structures.

3. For **Configure Routine Cache**, you can accept the default **Automatic** setting to allocate an amount of memory equal to 10% of the database cache for the default 8-kilobyte buffer size, or select **Specify Amount** and enter the routine cache memory allocation, in megabytes.

For more information about specifying the database and routine caches, see the globals and routines parameters in the *Configuration Parameter File Reference*.

2.1.2 Setting the Maximum Per-Process Memory

The **Maximum Per-Process Memory (KB)** setting on the Memory and Startup page specifies the maximum memory that can be allocated to a new process running on this InterSystems IRIS instance. The allowed range is 256 KB to 2147483647 KB; the default is the initial value of the bbsiz parameter. InterSystems recommends setting this equal to -1 (which resolves to the maximum value) for most circumstances.

Note: It is not necessary to reset this value unless you have set it lower than its default. If you receive <STORE> errors, increase the size and restart the process that created the error.

Process private memory, which is used for symbol table allocation and various other memory requirements (for example I/O device access structures and buffers), is allocated in increasing extents as required by the application until the specified maximum is reached. Once private memory is allocated to a process, it is not deallocated until the process exits.

See bbsiz for more information about this setting, including how to change it programmatically. For a detailed discussion of process memory in InterSystems IRIS, see Process Memory in InterSystems Products.

2.1.3 Configuring Shared Memory Heap (gmheap)

The shared memory heap is the memory used by InterSystems IRIS for purposes other than the routine and database caches. This setting is not on the Memory and Startup page. You can configure **gmheap** on the **Advanced Memory** page (**System Administration** > **Configuration** > **Additional Settings** > **Advanced Memory**) or programmatically; for more information, see gmheap in *Configuration Parameter File Reference*.

To see details of used and available memory for **gmheap**, navigate to the Shared Memory Heap Usage page (**System Operation > System Usage**) and click the **Shared Memory Heap Usage** link; for more information, see Shared Memory Heap Usage.

If your application will utilize a large number of SQL queries or if you plan on enabling parallel dejournaling, you will need to allocate additional memory to `gmheap`. For more information, see Shared Memory Considerations and System Requirements for Parallel Dejournaling.

2.2 IPv6 Support

You can enable or disable the use of IPv6 addresses in InterSystems IRIS by navigating to the **Startup Settings** page (**System Administration** > **Configuration** > **Additional Settings** > **Startup**) page; in the **IPv6** row, click **Edit**. Select **IPv6** to enable this option.

Note: This option is visible only if the network to which this InterSystems IRIS instance is connected permits IPv6 addressing.

When IPv6 is enabled, InterSystems IRIS accepts IPv6 addresses, IPv4 addresses, or DNS forms of addressing (host names, with or without domain qualifiers); when IPv6 is disabled, InterSystems IRIS accepts only IPv4 addresses or DNS forms of addressing.

When dotted-decimal IPv4 addresses (for example, 192.29.233.19) are specified, an IPv4 connection is attempted; when colon-separated IPv6 addresses (for example, 2001:fecd:ba23:cd1f:dcb1:1010:9234:4085) are specified, an IPv6 connection is attempted. When a DNS name (for example, mycomputer.myorg.com) is specified, it resolves to an actual IP address: first, it attempts to make an IPv4 connection; then, if an IPv4 connection cannot be made, it attempts an IPv6 connection.

InterSystems IRIS allows Internet addresses to be supplied in DNS, IPv4 and IPv6 formats. For example, localhost, 127.0.0.1, and ::1 are representations of the loopback address in each format, respectively. Detailed information about IPv6 addressing can be found in the following Internet Engineering Task Force documents:

- IP Version 6 Addressing Architecture (RFC 4291)
- Application Aspects of IPv6 Transition (RFC 4038)
- Format for Literal IPv6 Addresses in URL's (RFC 2732)

IPv6 addressing can also be checked and controlled using the **IPv6Format** method of the %SYSTEM.Process class (for the current process) or the **IPv6** method of the Config.Startup class (for the system generally).

Even though an InterSystems IRIS instance may be using an IPv4 network, IPv6 addresses can still be used as input to the various services provided that the IPv6 address supplied has a valid IPv4 equivalent. The loopback address used earlier in this section is such an example; *RFC 4291* describes several more formats. Thus, the various InterSystems services will accept either IPv4 or IPv6 addresses without error as long as the address form given can be validly converted for use on the connected network. So all of these forms (and several more) are acceptable

- localhost (DNS)
- 127.0.0.1 (IPv4)
- ::FFFF:127.0.0.1 (IPv4 mapped IPv6 format)
- 0:0:0:0:0:0:0:1 (full IPV6)
- ::1 (compressed IPv6)

as valid representations of the loopback address.

Generally, when asked for an Internet address that has been supplied to an InterSystems service earlier, InterSystems IRIS does not alter the address format. Addresses supplied in IPv4, or IPv6 format are returned as IPv4 or IPv6, respectively. The only exception is that addresses supplied as host names and translated by the Domain Name Server (DNS) may be returned in whatever form the DNS returns.

Note: InterSystems IRIS does not support the use of wildcard characters or ranges in IPv6 addresses.

3

Configuring Namespaces

This page provides information on configuring namespaces in InterSystems IRIS® data platform. For background information, see Namespaces and Databases. Also see Adding Mappings to a Namespace.

You can configure namespaces on the **Namespace** page of the Management Portal, which you can navigate to by selecting **System Administration** on the home page, then **Configuration**, then **System Configuration**, then **Namespaces**.

The absolute limit to the number of namespaces within a single InterSystems IRIS instance is 2048. The size of the namespaces table is not configurable.

3.1 Rules for Namespace Names

The following rules apply to a namespace name:

- It must be between one and 64 characters long.
- It must start with an alphabetic character or a percent sign (%); the remainder can include alphanumeric characters, dashes, or underscores.

For information on system-supplied namespaces, see System-Supplied Namespaces.

3.2 Create/Modify a Namespace

To create a namespace, click **Create New Namespace** to display the **New Namespace** page, then do the following:

1. Enter a **Name** for the namespace. See Rules for Namespace Names.
2. You can **Copy from** an existing namespace, creating a duplicate of the selected namespace. In this case, all other options will be made unavailable except for the **Web application** check box described in step 6 below.
3. Choose whether the default database for globals is **local** or **remote**.
4. **Select an existing database for Globals** for the default Global mapping of this namespace or click **Create New Database**, which launches either the database wizard or the remote database wizard.
5. Optionally, you can choose whether the default database for routines is **local** or **remote**, then either use the **Select an existing database for Routines** drop-down to choose a database for the default Routine mapping of this namespace, or click **Create New Database**, which launches either the database wizard or the remote database wizard.

6. Select the **Create a default Web application for this namespace** check box if you are creating a web application that accesses this namespace.
7. Optionally, you can clear the Enable namespace for interoperability productions check box. For more information, see Create an Interoperability-Enabled Namespace below.
8. After entering the required information, click **Save** to add your namespace to the configuration.

3.2.1 Create an Interoperability-Enabled Namespace

When you create a namespace on an InterSystems IRIS instance, the **Enable namespace for interoperability productions** check box is displayed at the bottom of the **New Namespace** page and is automatically selected. To create a namespace that is not interoperability-enabled, clear this check box before clicking **Save**.

If you do not clear the check box and create an interoperability-enabled namespace, the system automatically performs additional configuration tasks for the new namespace, as follows:

- If the default globals database for this namespace is an existing database, it upgrades and recompiles some classes in that database.

  CAUTION: If you are also using this database in other namespaces, you might consider this change undesirable. When you create a new namespace in an InterSystems IRIS instance, carefully consider whether it is appropriate for this namespace to reuse an existing database.

- It defines global mappings, routine mappings, and package mappings that make the InterSystems IRIS system classes and data available to the namespace.
- It adds nodes to the *^%SYS* global.
- It creates a web application for the namespace, using the application name required by InterSystems IRIS: `/csp/`*namespace*.

3.2.2 Create a Foundation Namespace for Healthcare Interoperability

To create a *Foundation* namespace and production for healthcare interoperability, use the InterSystems IRIS for Health™ Installer Wizard or the HealthShare® HealthConnect Installer Wizard, rather than the **New Namespace** page. Once you have activated a Foundation namespace in the Installer Wizard, if you later need to modify an option, such as the routine database, you can use the Namespaces page.

3.3 Rename a Namespace or Modify Default Mappings

You can rename a namespace, or change the databases to which your namespace is mapped without restarting InterSystems IRIS, using the following procedure:

1. Go to the **Namespaces** page (**System Administration** > **Configuration** > **System Configuration** > **Namespaces**).
2. On the **Namespaces** page, click the name of namespace you wish to modify.
3. Change or replace the existing name to rename the namespace.

Important: If you are renaming an interoperability-enabled namespace, you must take additional steps to complete the process.

a. Open the InterSystems Terminal from the System Tray.

b. Enter:

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

Where <*NewNamespace*> is the new name of the existing namespace.

c. Go to the **Web Applications** page (**System Administration** > **Security** > **Applications** > **Web Applications**).

d. Find the name of the application that corresponds to the old name of the namespace, and click **Delete**.

e. Click the name of the application that corresponds to the new name of the namespace.

f. Select **Namespace Default Application** and click **Save**.

g. In the Terminal, enter:

```
do ##class(%EnsembleMgr).DisableNamespace("<OldName>",1)
```

Where <*OldName*> is the original name of the namespace that you are renaming.

4. Choose the **Default Database for Globals**, the **Default Database for Routines**, and the **Default Database for Temporary Storage** from the list of defined databases.

   Note: Selecting a database that is configured not to journal globals (that is, the **Journal globals** property is set to **No**) from the **Default Database for Temporary Storage** drop-down list is not the same as selecting IRISTEMP; for more information, see Using Temporary Globals and IRISTEMP.

5. Click **Save**.

Users directly accessing the database at the time of the change may need to log out of and then back into InterSystems IRIS to update their namespace mappings.

3.4 Delete a Namespace

You can delete a namespace, including all mappings associated with it:

1. Navigate to the **Namespaces** page (**System Administration** > **Configuration** > **System Configuration** > **Namespaces**) and click **Delete** in the row of the namespace you want to delete.
2. On the **Delete Namespaces** page, if you want to delete the Web Gateway pages from the physical path, select the check box.
3. To delete the namespace and associated mappings, click **Perform Action Now**.

3.5 Programmatically Configure a Namespace

InterSystems IRIS also provides programmatic tools that are useful for automating namespace configuration. You can use the Config.Namespaces class to create and configure namespaces; see the *InterSystems Class Reference* for more information.

Another method for configuring namespaces is to include the CreateNamespace, ModifyNamespace, or DeleteNamespace operations in conjunction with configuration merge. Configuration merge allows you to customize an InterSystems IRIS instance by applying a declarative merge file, which specifies settings and operations to apply to that instance. For more information about this feature, see Using Configuration Merge to Deploy and Reconfigure InterSystems IRIS.

3.6 See Also

- Namespaces and Databases (which includes information on system-defined namespaces and databases)
- Adding Mappings to a Namespace
- Using Configuration Merge to Deploy and Reconfigure InterSystems IRIS

4

Adding Mappings to a Namespace

As part of configuring a namespace, you can also map globals, routines, and class packages from other databases on the same or different systems.

This allows simple references to data which can exist anywhere and is the primary feature of a namespace. You can map whole globals or pieces of globals; this feature allows data to easily span disks.

Note: Mappings are sorted alphabetically; if subscripts are specified, they are sorted by name and subscript. For more information, see Formal Rules about Globals.

Important: If there is mapped content with the same identifier as local content (such as a package, class, global, or routine name), the mapped content will be visible, rather than the local content. As such, you should be as specific with your mappings as possible, to prevent mapping away from existing content.

Be sure to follow Rules and Guidelines for Identifiers when naming content and creating mappings, to avoid any unwanted conflicts.

4.1 Global Mappings

You can add a mapping for a new global to your namespace at the global and global subscript level that overrides the default database mapping for globals of the namespace:

1. Navigate to the **Namespaces** page (**System Administration** > **Configuration** > **System Configuration** > **Namespaces**) and click **Global Mappings** in the row of the namespace where you want to map the global.

2. From the **Global Mappings** page click **New**.

3. Select the **Global database location** database where the global is located.

4. Enter the **Global name**. You can use the `*` character as part of the global name to specify multiple globals, for example **`ABC*`**.

5. Enter the **Global subscripts to be mapped**. The subscript reference must begin with an open parenthesis. Some examples follow:

   ```
   (1)
   ("A")
   (1):(5)
   ("A"):("Z")
   ("B",23,"m"):("E",5)
   (BEGIN):("X")
   ("Y"):(END)
   ```

Note: When specifying a range (for example, `("A"):("Z")`, the range is “from-to” (not “from-through”) the specified subscripts; that is, the lower end of a defined subscript range is inclusive, while the upper end of the defined subscript range is exclusive. For example, `Name (1):(10)` includes `Name (1)` but does not include `Name (10)`; the exclusive upper range allows you to have a defined upper boundary when working with subscripted ranges, such as `Name ("a"):("b")`, where `Name ("aa")` and `Name ("aaaaa")` are equally valid ranges to precede `Name ("b")`.

You can use the reserved words `BEGIN` and `END` to refer to the first and last possible subscripts; however, you cannot use the asterisk (`*`) wildcard with subscripted globals because global subscripts must be mapped individually.

For more information about subscript-level mapping (SLM) ranges, see Setting Up Global Mappings.

6. Click **Advanced** to display the following:
   a. Select the **Collation**. Collation applies only to new subscript-level mapping globals.
   b. Select the **Lock Database Location**. For more information see Global in the [Map] section of the *Configuration Parameter File Reference*.
7. Click **OK**.

   Note: **>>** displayed in the first column of the new mappings row indicates that you opened the mapping for editing.

8. To save the mappings in the cpf file, click **Save Changes**.

Important: While it is possible to add a mapping that changes the database location of an existing global, this does not actually move the global. As a consequence, the global becomes inaccessible, as it remains in the original database while the namespace expects to find it in the newly mapped database. For a new mapping for an existing global to be successful, you must relocate the global manually, for example using the Terminal or the Management Portal, by creating it on the new database and removing it from the original database.

4.2 Routine Mappings

You can add mappings to your namespace at the routine level that overrides the default database mapping for routines of the namespace:

1. Navigate to the **Namespaces** page (**System Administration** > **Configuration** > **System Configuration** > **Namespaces**) and click **Routine Mappings** in the row of the namespace where you want to map the global.
2. From the **Routine Mappings** page, click **New**.
3. Select the **Routine database location** database where the routine is located.
4. Enter the **Routine name**. The routine does not have to exist when you map it (that is, it can be the name of a routine you plan to create).
5. Click **OK**.

   Note: **>>** displayed in the first column of the new mappings row indicates that you opened the mapping for editing.

6. To save the mappings in the cpf file, click **Save Changes**.

For example, using the preceding Test Namespace Mapping example, if you plan to create a schedule routine (for example, BOSZZairline) in the airports database (in the FlightSchedule namespace) and you want it to be available to users in the TravelAgent namespace, navigate to the **Routine Mappings** page (in the TravelAgent namespace row), then click **New Routine Mapping**. Enter the information as shown in the following **Routine Mapping** dialog box:

Important: When you map one or more routines, be sure to identify all the code and data needed by those routines, and ensure that all that code and data is available in all the target namespaces. The mapped routines could depend on the following items:

- Include files
- Other routines
- Classes
- Tables
- Globals

Use additional routine, package, and global mappings as needed to ensure that these items are available in the target namespaces.

4.3 Package Mappings

You can add a class package mappings which makes all the classes within a package (and all the generated routines for those classes) in a specific database visible to another namespace:

1. Navigate to the **Namespaces** page (**System Administration** > **Configuration** > **System Configuration** > **Namespaces**) and click **Package Mappings** in the row of the namespace where you want to map the package.
2. From the **Package Mappings** page, click **New**.
3. Select the **Package database location** database where the package is located.
4. Select the **Package name**. The package does not have to exist when you map it (that is, it can be the name of a package you plan to create); you can specify a new package name, as follows:
   a. Click **New Package**.
   b. In the **New package name** text box, enter a name.
5. Click **OK**.

   Note: **>>** displayed in the first column of the new mappings row indicates that you opened the mapping for editing.
6. To save the mappings in the cpf file, click **Save Changes**.

See Package Mapping for a description of packages and the procedure for mapping them.

Important: When you map a package, be sure to identify all the code and data needed by the classes in that package, and ensure that all that code and data is available in all the target namespaces. The mapped classes could depend on the following items:

- Include files
- Routines
- Other classes
- Tables
- Globals

Use additional routine, package, and global mappings as needed to ensure that these items are available in the target namespaces.

InterSystems IRIS also provides functionality to make a package available in all namespaces through a single action.

4.4 Mapping to All Namespaces

In addition to mapping globals, routines, and packages to specific namespaces, you can map them to all namespaces. To enable this form of mapping:

1. First, create a namespace named %ALL, as described in Create/Modify a Namespace.

   Note: %ALL is not visible except for the purposes of mapping data; that is, it is not a real namespace, but a mechanism for mapping data to all namespaces.

2. Then, make the desired mappings in the %ALL namespace, as described in the following topics:
   - To map globals, see Global Mappings.
   - To map routines, see Routine Mappings.
   - To map packages, see Package Mappings.

These %ALL mappings apply in all namespaces. You cannot create namespace-specific mappings to resources that are mapped in the %ALL namespace, as %ALL mappings override any namespace-specific mappings to the same resource.

CAUTION: %ALL mappings apply to every namespace, including %SYS. It is possible to break certain features by creating a mapping that overrides a routine or global the instance relies on.

You should create mappings as narrowly as possible when using %ALL. Additionally, closely follow the Rules and Guidelines for Identifiers; in particular, do not create %ALL mappings for any globals listed in Global Names to Avoid.

When you create a subscript-level mapping in the %ALL namespace, a mapping for the root global is automatically created to %DEFAULTDB. The %DEFAULTDB variable represents the default database for any given namespace.

4.5 See Also

- Namespaces and Databases

- Configuring Namespaces
- Using Configuration Merge to Deploy and Reconfigure InterSystems IRIS

5

Configuring Local Databases

This page provides information on configuring local databases in InterSystems IRIS® data platform. Note that you can make most database configuration changes dynamically; you can create and delete databases and modify database attributes while the system is running.

5.1 Database Considerations

For background information, see Namespaces and Databases. This section presents additional concepts relevant to system administrators.

CAUTION: On Windows systems, *do not* use file compression on IRIS.DAT database files. (Files are compressed by right-clicking a file or folder in Windows Explorer and selecting **Properties**, then **Advanced**, then **Compress contents to save disk space**; once compressed, a folder name or filename is rendered in blue in Windows Explorer.) If you compress a IRIS.DAT file, the instance to which it belongs will fail to start, with misleading errors.

5.1.1 Database Size and Growth

InterSystems IRIS databases dynamically expand as needed (assuming free space is available), though you can specify a maximum size. A database can grow until it is 32 terabytes if you are using the default 8KB block size.

5.1.2 The Database Cache

Each InterSystems IRIS system maintains a database cache — a local, shared memory buffer used to cache data retrieved from the physical databases. This cache greatly reduces the amount of costly I/O operations required to access data and provides much of the performance benefits of InterSystems IRIS. (For information about allocating the database cache, see Memory and Startup Settings.)

5.1.3 Database Total Limit

The absolute limit on the number of databases that can be configured within a single InterSystems IRIS instance (given sufficient storage space) is 15,998. There are other limitations as follows:

- The directory information for databases cannot exceed 256 KB. This means that if the average length of a database directory name is longer, an instance can have fewer total databases. The following formula describes this relation:

  $maximum\_DBs = 258048/ (avg\_DB\_path\_length + 3)$

For example, if all database directory paths are of the form `c:\InterSystems\IRIS\mgr\DB`*NNNN*`\`, the average length is 33 bytes. Thus, the maximum number of databases is 7,168, calculated as follows: 258048/ (33 + 3) = 7168.

- Mirrored databases count twice toward the absolute limit of 15,998. If all databases on the instance are mirrored, the effective limit is 7,499 databases. This is because InterSystems IRIS creates two database definitions for mirrored databases; one for the directory path (c:\InterSystems\IRIS\mgr\DBNNNN\), and one for the mirror definition (:mirror:MIRRORNAME:MirrorDBName).
- The number of databases that can be in use simultaneously is limited by the operating system's restriction on the number of open files (either per process or system-wide). InterSystems IRIS reserves approximately half of the operating system's open file allocation for its own use and devices.

5.1.4 Database Configuration Considerations

The following are tips to consider to consider when configuring databases:

- InterSystems IRIS provides a seamless option to spread data across multiple physical database (`IRIS.DAT`) files. Therefore, you can build applications with multiple databases or splitting data by global or subscript-level mappings, as appropriate.
- Keep database sizes within a manageable value based on the infrastructure available for administration tasks such as backup, restore, integrity checks, etc.
- It is recommended that stream globals (if storing streams within `IRIS.DAT` database files) be global mapped to a separate database, and that the stream database(s) be configured with a large (64 KB) block size.
- Depending on your workload, it may be beneficial to consider alternate (larger) block sizes than the default 8 KB database block size. For general guidelines, see Large Block Size Considerations below.

5.1.5 Large Block Size Considerations

In addition to the 8 KB (default) block size supported by InterSystems IRIS (which is always enabled), you can also enable the following block sizes:

- 16 KB (16384)
- 32 KB (32768)
- 64 KB (65536)

However, you should exercise caution when creating your database to use large block size because using them can impact the performance of the system. Consider the following before enabling and using large block sizes:

- If your application workload consists primarily of sequential inserts or sequential reads/queries, large block sizes may improve performance.
- If your application workload consists primarily of random inserts or random reads/queries, large block sizes may degrade performance. Since larger block sizes result in fewer blocks being cached for a given total size of database cache, to reduce the impact on random database access, you should also consider making more total memory available as database cache.
- For index-type databases, the default block size (8 KB) ensures optimum performance; larger block sizes potentially degrade performance. If you are considering larger block sizes for your data, you should consider mapping index globals to a separate 8 KB block size database.

To create a database that uses block sizes other than the supported blocks, do the following:

1. Enable the block sizes using the setting on the **Startup Settings** page (**System Administration** > **Additional Settings** > **Startup**); see DBSizesAllowed.
2. Configure the database cache for the enabled block size on the **Startup Settings** page (**System Administration** > **Additional Settings** > **Startup**), as described in Memory and Startup Settings.
3. Restart InterSystems IRIS.
4. Create the database as described in Create Local Databases.

5.1.6 Database Compatibility Considerations

As described in the Create a Local Database procedure, you can copy or move an InterSystems IRIS database to an instance other than the one in which it was created by copying or moving its IRIS.DAT file, or temporarily mount a database created in another instance on the same system. You can also restore a backup of a database (see Backup and Restore) to an instance other than its original instance. To avoid data incompatibility, however, the following requirements must be met:

- The target (new) instance must use the same character width (8-bit or Unicode; see Character Width Setting) and the same locale (see Using the NLS Settings Page of the Management Portal) as the instance that created the database.

  The one exception to this requirement is that an 8-bit instance using a locale based on the ISO 8859 Latin-1 character set is compatible with a Unicode instance using the corresponding wide character locale. For example, a database created in an 8-bit instance using the **enu8** locale can be used in a Unicode instance using the **enuw** locale.

- If the source and target instances are on systems of different endianness, the database must be converted to the endianness of the target instance before being used.

  Depending on the platform, multibyte data is stored with either the most significant byte or the least significant byte in the lowest memory address (that is, first): when the most significant byte is stored first, it is referred to as “Big-endian;” when the least significant byte is stored first, it is referred to as “Little-endian.”

  When defining a database using an existing IRIS.DAT created on a system of different endianness, use the cvendian utility (see Using cvendian to Convert Between Big-endian and Little-endian Systems) to convert the database before you use it. When restoring a backup of a database to a system of different endianness than the source system, see Considering Endianness.

5.2 Rules for Database Names

The following rules apply to a database name:

- It must be between one and 64 characters long
- It must start with an alphabetic character or an underscore; the remainder can include alphanumeric characters, dashes, or underscores

For information on system-supplied databases, see System-Supplied Databases.

5.3 Viewing Local Databases

To view the local databases, navigate to the **Local Databases** page (**System Administration** > **Configuration** > **System Configuration** > **Local Databases**).

The **Local Databases** page displays the following information about the databases on your system:

- **Name** — Database name.
- **Mirror** — If the database is mirrored, the name of the mirror; for more information see Add Databases to the Mirror.
- **Directory** — Location of the IRIS.DAT file.
- **Size (MB)** — Size of the database in megabytes.
- **Status** — Specifies whether or not the database is mounted, unmounted, or dismounted; if it is mounted, specifies whether it has read-only or read-write permission. For more information, see "The Local Databases List Information" table in Maintaining Local Databases.
- **Resource Name** — The name of the database resource that governs access to the database; for more information, see Using Resources to Protect Assets.
- **Encrypted** — Specifies whether or not the database is encrypted; for more information, see Encryption Guide.
- **Journal** Specifies whether or not the database is journaled; for more information, see Journaling.

You can use this page to:

- Create a Local Database
- Edit a Local Database's properties
- Relocate a Local Database
- Delete a Local Database

5.4 Creating a Local Database

To create a local database:

1. Navigate to the **Local Databases** page (**System Administration** > **Configuration** > **System Configuration** > **Local Databases**).
2. Click **Create New Database** to open the **Database Wizard**.
3. Enter a database name in the text box. A database name must not already be in use within the InterSystems IRIS instance; also see Rules for Database Names.
4. The first time you create a local database in an InterSystems IRIS instance using a particular browser, you must either
    - Enter the name of the database directory, in which case this directory, containing the IRIS.DAT file, is created in c:\InterSystems\mgr once you confirm it.
    - Click the folder icon to browse to an existing directory, in which case the IRIS.DAT file is created in that directory.

    Thereafter, by default a directory of the same name as the database name you provide, containing the IRIS.DAT file, is created in the same location as the previous database directory. For example, if you first create database db22 in any directory under c:\InterSystems\mgr, when you click **Create New Database** again and enter db33 in the **Enter the name of your database** box, c:\InterSystems\mgr\db33 is automatically filled into the **Database directory** text box. If you change this to c:\InterSystems\db33 and create db33, the base directory c:\InterSystems will be filled in the next time.

    Note: InterSystems does not support the use of symbolic links when configuring database directories.

5. Click **Next** to continue configuring the database. If a IRIS.DAT file already exists in the directory you specified, you are warned of this and can either

  - Click **Finish** to use the existing file, in which case all of the databases characteristics are determined by the IRIS.DAT file. You would typically do this when copying or moving a database from another instance, or temporarily mounting a database created in another instance on the same system.
  - Click **Back** to specify another directory, then click **Next** again to continue specifying the characteristics of the new database in the next step.

6. In the **Initial Size** text box, type the number of megabytes for your database size (the default is 1 MB).

   Note: You cannot create or edit a database so that its size is larger than the total available disk space. If the size you specify is within 90% of the disk's free space, you are warned and must confirm the action.

7. Select the desired block size from the **Block size for this database will be** drop-down list. By default, all new databases are created with a **Block Size** of 8 KB.

   CAUTION: Do not select block sizes other than 8 KB from the drop-down list unless you have read and understand the guidelines described in Large Block Size Considerations.

8. Select whether or not you want to journal globals in this database from the **Journal globals?** drop-down list. See Journaling.

   Note: If you are configuring the database to store temporary globals, setting the **Journal globals** property to **No** is not the same as storing the temporary globals in IRISTEMP; for more information, see Using Temporary Globals and IRISTEMP.

9. If encryption is activated, you can encrypt this database by selecting **Yes** for **Encrypt Database?**.
10. If the instance is part of a mirror, you can add this database to the mirror by selecting **Yes** for **Mirrored Database?**. See Add Databases to the Mirror for information about creating mirrored databases.
11. From this panel onward, you can click **Next**. to continue configuring the database or Finish to accept the remaining defaults
12. Choose the resource to control access to this database:
    - Default — %DB_%DEFAULT
    - Existing — Choose from a list of existing database resources
    - New — Create a new database resource (the new name defaults to %DB_% *database name*)
13. Click **Next** to view a list of the database attributes.
14. Click **Finish** to add your database.

You are now ready to configure and manage your new database.

Note: To protect you from accidentally corrupting a database, you cannot open or write to an operating system file called IRIS.DAT, even if it is not a mounted database.

5.5 Editing a Local Database’s Properties

To edit a local database’s properties:

1. Navigate to the **Local Databases** page (**System Administration** > **Configuration** > **System Configuration** > **Local Databases**).

2. Click the name of the database.

   The information displayed varies depending on whether or not the database is mirrored. See the sections for:

   - Nonmirrored Local Databases
   - Mirrored Local Databases

5.5.1 Editing Nonmirrored Local Database Properties

If you are editing a nonmirrored database, the system displays the following properties (Create a Local Database describes many of these fields).

- **Name**
- **Directory** (this setting must always reflect the location of the IRIS.DAT database file)
- **Encrypted** (cannot be changed)
- **Mirrored** — Click **Add to Mirror***mirror_name* to add the database to the mirror in which the InterSystems IRIS instance is the primary failover member. (This option is available only when the instance is the primary in a mirror.) See Add an Existing Database to the Mirror.
- **Block Size (Bytes)** (cannot be changed)
- **Size (MB)** — There are three size settings, as follows:
  - Change **Current** to modify the current size of the database.

    Note: You cannot create or edit a database so that its size is larger than the total available disk space. If the size you specify is within 90% of the disk's free space, you are warned and must confirm the action.

  - **Expansion** sets the amount by which to expand the database when required; the default (and recommended) setting of zero (0) indicates 12% of current size or 10 MB, whichever is larger. When using 12% of the current size, the expansion size will not be greater than 1GB.
  - **Maximum** specifies the maximum size to which the database can grow, in megabytes; the default setting of zero (0) indicates no maximum. To modify this setting, you can enter a new number of MB, or you can precede a number by **+** or **-**, for example **+10** or **-20**, to enlarge or reduce the maximum by the specified amount. When you reduce the maximum size of a database, you are warned and must confirm the action. Note, the IRISAUDIT database must be configured to have no maximum size.
- **Resource Name** — Select the resource with which to associate the database. Click the resource icon next to the drop-down to display the Resources page so you can create a resource.
- **New Global** — Specify attributes for new globals.
- **Global Journal State** — Select to enable journaling, clear to disable. See Journaling.
- **Preserve global attributes on delete** — Specify whether a global’s directory entry and attributes should be preserved when it is deleted; attributes include collation, journaling status, and growth pointer. Select to preserve a global’s directory entry and attributes when the global is entirely deleted; clear to remove the directory entry and attributes.
- **Mount Read-Only** — Select to specify that the database be mounted as read-only; clear to specify that it be mounted as read-write.
- **Mount Required at Startup** — Select to indicate that the database must be mounted when InterSystems IRIS starts up; if the database cannot be mounted, InterSystems IRIS does not start. This lets you ensure that journal recovery and transaction rollback can be performed on the database before startup following a crash (as described in Journaling). Clear to let InterSystems IRIS start without first mounting the database.

> **Note:** By default, this setting is selected for required InterSystems IRIS databases (for example, IRISLIB and IRISAUDIT) and cannot be changed. The default is cleared but can be selected for databases that you create, as well as the USER and ENSLIB databases). For additional information about database status and explicitly dismounting and mounting databases, see "The Local Databases List Information" table in Maintaining Local Databases.

- **Stream Location** — Click the **Browse** button to select the directory in which streams associated with this database are stored. By default, the stream location for a local database is a subdirectory named stream in the database **Directory**, which is one of the above fields (for example, *install-dir*\mgr\DB1\stream).

  Note: InterSystems recommends that you use the default location.

5.5.2 Editing Mirrored Local Database Properties

If you are editing a mirrored database, the system displays the following properties (Create a Local Database describes many of these fields).

Note: Journaling is required for a mirrored database, therefore the **Global Journal State** setting does not appear.

- **Name**
- **Mirror Name** — Name by which the database is identified within the mirror; cannot be changed.
- **Directory** (this setting must always reflect the location of the IRIS.DAT database file)
- **Encrypted** (cannot be changed)
- **Stream Location** — Click the **Browse** button to select the directory in which streams associated with this database are stored. By default, the stream location for a local database is a subdirectory named stream in the database **Directory**, which is one of the above fields (for example, *install-dir*\mgr\DB1\stream).

  Note: Like other database-related data that is not contained in the database itself (see Mirror Configuration Guidelines), a mirrored database's file streams are *not* mirrored. (For information about file streams, see Working with Streams.)

  InterSystems recommends that you use the default location.

- **Resource Name** — Select the resource with which to associate the database. Click the resource icon next to the drop-down to display the Resources page so you can create a resource.
- **Block Size (Bytes)** (cannot be changed)
- **Collation** — Among global attributes, only the collation attribute can be changed, for new globals only.
- **Preserve global attributes on delete** — Specify whether a global's directory entry and attributes should be preserved when it is deleted; attributes include collation, journaling status, and growth pointer. Select to preserve a global's directory entry and attributes when the global is entirely deleted; clear to remove the directory entry and attributes.
- **Mount Read-Only** — Select to specify that the database be mounted as read-only; clear to specify that it be mounted as read-write.
- **Mount Required at Startup** — Select to indicate that the database must be mounted when InterSystems IRIS starts up or becomes a mirror primary; if the database cannot be mounted, InterSystems IRIS does not start or become primary. This lets you ensure that journal recovery and transaction rollback can be performed on the database before startup following a crash (as described in Journaling) and that open transactions on the former primary have been rolled back as part of failover. Clear to let InterSystems IRIS start without first mounting the database. For additional information

about the **Mount Required at Startup** setting, see “The Local Databases List Information” table in Maintaining Local Databases.

- **Local Properties** — This area contains three size settings, as follows:
  - Change **Size** to modify the current size of the database.

    Note: You cannot create or edit a database so that its size is larger than the total available disk space. If the size you specify is within 90% of the disk's free space, you are warned and must confirm the action.

  - **Expansion** sets the amount by which to expand the database when required (and assuming free space is available); the default (and recommended) setting of zero (0) indicates 12% of current size or 10 MB, whichever is larger.
  - **Maximum** specifies the maximum size to which the database can grow, in megabytes; the default setting of zero (0) indicates no maximum. To modify this setting, you can enter a new number of MB, or you can precede a number by **+** or **-**, for example **+10** or **-20**, to enlarge or reduce the maximum by the specified amount. When you reduce the maximum size of a database, you are warned and must confirm the action. Note, the IRISAUDIT database must be configured to have no maximum size.

  This area also contains the current, expansion, and maximum size settings for **Other System** — if the current instance is a failover member, this is the other failover member; if the current instance is an async member, this is the first failover member that the async could obtain the information from. For important information about how the properties of a mirrored database on the backup and async mirror members are synchronized with those on the primary, see Mirrored Database Considerations.

5.6 Relocating a Local Database

To move a local database’s IRIS.DAT file to a different directory, do the following:

1. Make note of the current database directory. You can view this information from the **Local Databases** page (**System Administration** > **Configuration** > **System Configuration** > **Local Databases**).
2. Perform a clean shut down of the instance, such as by using the **iris stop** command.
3. Copy the IRIS.DAT file and the stream directory from the current database directory to the desired location.

   Important: If there is an iris.lck file in the database directory, do not attempt to move the database. This means the database is still in use, and attempting to move it could lead to unforeseen problems. Contact the InterSystems Worldwide Response Center (WRC) for assistance.

   Note: InterSystems does not support the use of symbolic links when configuring database directories, and recommends choosing a fully resolved physical path for the new database location.

4. Open the iris.cpf file for the instance in a text editor. This file is usually located in the *installation directory*.
5. Locate the name of the database beneath the `[Databases]` section. Replace the old directory path with the new path, then save iris.cpf.
6. Delete the old database directory.
7. Start the InterSystems IRIS instance, and view the **Local Database** page to confirm that the directory is set to the new location.
8. If the database is mirrored, use **^MIRROR** to remove the original location of the mirrored database. Specifically use the option **Mirror management** > **Remove Mirrored Database**.

Then activate and catch up the mirror.

Important: After you relocate a local database directory, you must also update any systems that access the database remotely (such as ECP application servers). On each remote system, update the remote database directory to the new location, as described in Configuring Remote Databases.

5.7 Deleting a Local Database

To delete a local database, click the **Delete** link in the appropriate row. The Delete Database page displays information about the database you are deleting, and lets you:

- Select the namespaces mapped to this database for deletion. You cannot delete a database if a namespace is mapped to it, so unless you select all of the listed namespaces you cannot delete the database.

  You cannot delete namespaces that are also mapped to other databases. When this is the case, a link is provided to take you to the Namespaces page, where you can modify the database mappings of the namespaces involved. After you delete all mappings to another database, that database will be removed from the list of databases you have to delete.

- You can choose to delete the database's IRIS.DAT file, *if and only if*:
  - No other databases use this IRIS.DAT file.
  - You have marked all namespaces mapped to the database for deletion.

  If these conditions are not met, you can still delete the database from the current configuration, but the IRIS.DAT file cannot be deleted.

- Confirm that you want to delete the database after reviewing information about it by clicking **Delete Database Now**.

If you cannot or chose not to delete the IRIS.DAT file, the database is still removed from the Databases section of the InterSystems IRIS parameters file and therefore from the list of local databases displayed by the Management Portal.

5.8 Programmatically Configuring a Database

InterSystems IRIS also contains programmatic tools that are useful for automating database configuration. You can use the Config.Databases class to create and configure databases; see the *InterSystems Class Reference* for more information.

You can also configure databases using the ^DATABASE command line utility.

Another method for configuring databases is to include the CreateDatabase, ModifyDatabase, or DeleteDatabase operations in conjunction with configuration merge. Configuration merge allows you to customize an InterSystems IRIS instance by applying a declarative merge file, which specifies settings and operations to apply to that instance. For more information about this feature, see Using Configuration Merge to Deploy and Reconfigure InterSystems IRIS.

5.9 See Also

- Namespaces and Databases
- Configuring Remote Databases

- ^DATABASE
- Using Configuration Merge to Deploy and Reconfigure InterSystems IRIS
- Using cvendian to Convert Between Big-endian and Little-endian Systems
- Maintaining Local Databases

6

Configuring Remote Databases

This page describes how to add and delete remote databases. A *remote database* is a database that is physically located on another server system, as opposed to a *local database* which is physically located on the local server system.

6.1 Adding a Remote Database

You can define a remote database on the local server if the database's host is configured on that server as a distributed cache data server. For instructions for adding a data server, see Configuring an Application Server.

To add a remote database, follow these steps:

1. Navigate to the **Remote Databases** page (**System Administration** > **Configuration** > **System Configuration** > **Remote Databases**) and click **Create Remote Database** to launch the wizard.
2. Select the data server that hosts the database from the **Remote Server** drop down.
3. Choose how you want to specify the remote database directory from the **Remote Directory** radio buttons:
    - **Select databases from a list** lets you choose from a drop-down list of database directories on the remote server. If the remote data server cannot currently be reached, the drop-down list is empty.
    - **Enter your own database specification** lets you enter the database directory directly, but the Portal does not validate your entry.
4. Enter a database name (its name on the local server; it does not need to match its name on the remote server); see Rules for Database Names. You have defined a remote database.
5. You can optionally specify the directory in which streams associated with this database are stored. By default, the stream location for a remote database is the InterSystems IRIS `Temp` directory (*install-dir*`\mgr\Temp`).

    Note: InterSystems recommends that you use the default location.

6. Click **Save** to configure the remote database.

You can click the **Edit** link for a remote database at any time to modify the remote database fields.

6.2 Deleting a Remote Database

To delete a remote database, click the **Delete** link in the appropriate row. The Delete Database page displays information about the database you are deleting, and lets you:

- Select the namespaces mapped to this database for deletion. You cannot delete a database if a namespace is mapped to it, so unless you select all of the listed namespaces you cannot delete the database.

  You cannot delete namespaces that are also mapped to other databases. When this is the case, a link is provided to take you to the Namespaces page, where you can modify the database mappings of the namespaces involved. After you delete all mappings to another database, that database will be removed from the list of databases you have to delete.
- Confirm that you want to delete the database after reviewing information about it by clicking **Delete Database Now**.

This action simply removes the database from the local instance’s remote database configuration; the actual database and its local configuration on its host are not affected.

6.3 See Also

- Configuring an Application Server
- Configuring Local Databases

7

Configuring Task Manager Email Settings

You can set the Task Manager to send email notification when a task completes, as described in Using the Task Manager. On the **Task Manager Email Settings** page (**System Administration** > **Configuration** > **Additional Settings** > **Task Manager Email**), you can configure the notification settings described on this page.

7.1 Email Configuration Settings

SMTP Server and Port

Address and port of your outgoing SMTP (Simple Mail Transfer Protocol) mail server

SSL Config

The SSL configuration to use if you want to encrypt the email using SSL/TLS. If there are no SSL configurations on the instance, or you want to create a new one, see Create or Edit a TLS Configuration. If you do not select an SSL configuration, SSL/TLS will not be used.

SMTP Auth User and Password

Required only for SMTP authentication to the SMTP server. See RFC 2554 for details. If you do not provide entries, SMTP username and password are set to NULL.

Sender

Required only for SMTP authentication to the SMTP server. See RFC 2554 for details.

Reply To

Email address to which the recipient should reply

Success Subject

The formatted subject line of a successful task message. See the section “Parameters for Subjects and Messages” below.

Success Message

The formatted message sent after a successful task runs

Failure Subject

The formatted subject line of a failed task message

Failure Message

The formatted message sent after a task fails

Note: You can also configure email settings programmatically through the %SYS.Task.Config class.

7.2 Parameters for Subjects and Messages

Format the information in the subject and message text boxes using the task parameters listed at the bottom of the web page and defined in the following table. The web page includes examples.

Task Parameter	Description
ID	Task ID
DESCRIPTION	Task description
NAME	Task name
LASTSTARTED	Last time the task started
LASTFINISHED	Last time the task finished
SCHEDULED	Last scheduled starting time
CURRENTDATE	Date the email sent
CURRENTTIME	Time the email sent
STATUS	Return value of the task: • If the task is successful, this returns `""`. • If the task results in a trapped error, this returns text from a %Status value. • If the task results in an untrapped error, this returns $ZERROR.
TASKCLASS	Task class used for this task; for example, %SYS.Task.IntegrityCheck for a database integrity check task,
ERROR	Error code if task failed
SUCCESS	Completed message if task ran successfully

8

Configuring National Language Support (NLS)

A national language locale defines the character set in which all textual data is encoded by InterSystems IRIS® data platform. The character set is 16-bit Unicode UCS-16.

Each locale contains a number of character tables used by InterSystems IRIS when displaying text, collating data (see Collation), converting between uppercase and lowercase letters, matching patterns, and so on. Each locale defines the table to be used for each of these purposes, as well as other details such as date, time, and number formats.

Each InterSystems IRIS instance uses a single current locale; this is determined when the instance is installed, but can be changed at any time. When you change the current locale, some or all of the locale tables used by InterSystems IRIS change.

Installing a new locale does not result in any data conversion, but rather changes how data is represented.

Installing a new locale should not be a frequent operation; it is intended mainly as an upgrade option or the means to correct an installation choice. Always remember that data conversion may be needed and that special attention should be given to global subscripts.

You cannot alter the system locales provided with InterSystems IRIS, which are overwritten when the instance is upgraded.

8.1 Using the NLS Pages of the Management Portal

The **National Language Settings** pages (**System Administration** > **Configuration** > **National Language Settings**) let you browse existing locales and tables as well as create custom locales. You can install a new current locale, load a new table into memory, and more the using the Management Portal. When you select **System Administration** > **Configuration** > **National Language Settings**, the following options are available in the right-hand column:

- Configured Defaults
- Locale Definitions
- Import Locale

8.1.1 Configured Defaults

The **Configured Defaults** page (**System Administration** > **Configuration** > **National Language Settings** > **Configured Defaults**) displays the locale table currently used by default for each purpose within InterSystems IRIS. When writing ObjectScript code or using some utilities, it is possible to specify a particular table for a given purpose; the default table isused when no table is specified.

Each table name is color-coded to show whether the setting was inherited from the current locale at installation or specified using the NLS class packages, as described in Using System Classes for National Language Support.

The configuration defaults are a property of the instance, not of the locale. Therefore, when the instance is upgraded, the default selections are preserved.

8.1.2 Locale Definitions

From the **Locale Definitions** page (**System Administration** > **Configuration** > **National Language Settings** > **Locale Definitions**), you can select a locale at the **Select a locale** drop-down and perform several actions. The drop-down is always set to the current locale when the page first displays.

- Using the **Use locale date/time/number formats for [current locale]** drop-down, indicate whether or not you want to use the date, time, and number formats specified by the current locale. Note that this always applies to the current locale, not a locale you have selected in the **Select a locale** drop-down but not yet installed.
- To view the details of a selected locale, click **Properties**. The next page displays the locale properties grouped into categories. For locales you have added, you can edit the fields and click **Save** to save these changes. You cannot edit the system locales provided with InterSystems IRIS. The properties are as follows:
  - **Basic Properties**
  - **Date, Time, and Number Formats**
  - **Internal Tables** — You have two options when editing the internal tables:
    - **Edit Tables** — You may select or delete a table from the list boxes by double clicking an item, or by selecting an item and then clicking **>** or **<** to move it from the appropriate list.

      Tables that require at least one entry are indicated by an asterisk (`*`); the other tables may be left empty.
    - **Edit Defaults** — You may choose the default from the values you enter in the **Edit Tables** function of the **Internal Tables** category.
  - **Input/Output Tables** — You can edit, add, or remove a table when choosing to edit this category.
    - To edit a table, click the table in the first list. The table name appears in the lower box. You can modify the values and click **Save**.
    - To remove a table, click the table in the first list. The table name appears in the lower box; click **Remove**. A confirmation box displays offering you the option to **Cancel** or **OK** the delete.
    - To add a table, click **Add**. The lower box has the **Table** field enabled and the **Remove** option disabled. You can enter a table name and enter the **Output to** and **Input from** fields.

    Click **Save** when you have made all your updates. If the save is successful, the updated list appears; otherwise, an appropriate error message displays.
  - **Input/Output Defaults**
  - **Strings**
- To take further actions, click the following buttons:
  - **Validate** — Validates the selected locale, displaying an error message if the locale cannot be validated. This is useful when creating custom locales.

- **Copy** — Creates a copy of the selected locale, which you can then customize. The name of the copy must contain four characters beginning with `y` and ending with `8` or `w`. The default description is `Copy of`*%locale*, where *%locale* is the selected locale name. When the copy is created, it is added to the **Select a locale** drop-down.
- **Export** — Exports a locale to an .xml file. For example, you might export a custom locale you created and import it on another instance using the Import Locale page. The default name is loc_%locale.xml, where *%locale* is the selected locale. In addition, you can include the path of the export file; if you do not specify the path, the default location is *install-dir*\mgr.
- **Install** — Installs the selected locale as the current locale for the instance. An initial validation occurs; if it fails, an error message displays, otherwise you can continue with installation.
- **Load Table** — Lets you load a table from the selected locale (the current locale or another) into memory from disk. Select a table type and then a table name from the list populated after you select the type. Click **OK** to load the table or **Cancel** to close the dialog box and return to the **Locale Definitions** page.
- **Delete** — Deletes a locale. You can delete only custom locales; the button is disabled when a system locale is selected. You cannot delete the current locale even if it is a custom locale. You must confirm deletion of the locale before proceeding.

8.1.3 Import Locale

From the **Import Locale** page (**System Administration** > **Configuration** > **National Language Settings** > **Import Locales or Tables**), you can import locales or tables. For example, you can import a custom locale exported (as described in the previous section) from another instance.

1. Select the **Import Type** > **Locale** is the default.
2. Enter a file name and click **Import**. The only valid file extensions are .xml and .goq.
3. A message displays indicating how many locales, tables, and subtables have been imported.

8.2 Using the NLS Class Packages

System Classes for National Language Support contains details on using both the %SYS.NLS and Config.NLS class packages.

The %SYS.NLS Classes section contains details on using the following classes:

- %SYS.NLS.Device — Properties of the current device.
- %SYS.NLS.Format — Date, time, and number formats.
- %SYS.NLS.Locale — Basic properties of current locale (read-only).
- %SYS.NLS.Table — System and process tables (I/O and internal).

The Config.NLS Classes section contains details on using the following classes:

- Config.NLS.Locales
- Config.NLS.SubTables
- Config.NLS.Tables

You can also find details on each of these classes in the *InterSystems Class Reference*.

9

Configuring Work Queue Manager Categories

The Work Queue Manager enables you to distribute work to multiple concurrent processes programmatically in order to improve performance. For more information, see Using the Work Queue Manager.

When your code initializes a set of worker jobs using the Work Queue Manager, that code can specify the category that supplies the worker jobs. A *category* is an independent pool of workers with its own maximum of workers.

The system supplies two categories that you cannot delete: Default and SQL. By default, the maximum number of workers for these categories is **Dynamic**, which is equivalent to twice the number of cores available to your system.

The **Work Queue Manager Categories** page enables you to create, modify, and delete categories as described in the following sections. Any changes that you make are reflected in the configuration parameter file, `iris.cpf`.

9.1 Creating a Work Queue Manager Category

From the **Work Queue Manager Categories** page, you can create new categories.

Category names must be unique and are case sensitive. Additionally, category names can include only letters, numbers, dashes, underscores, and periods, and have a maximum of 64 characters.

To create a new category, do the following:

1. Navigate to **System Administration** > **Configuration** > **System Configuration** > **WQM Categories**.
2. Click **Create Category**.
3. In the **Category name** field, type a name for the category.
4. In the **Max active workers** field, select the maximum number of active worker jobs kept in the pool of jobs that service requests in this category. InterSystems IRIS detects idle jobs and automatically starts new jobs to keep the maximum active job number around this limit. By default, InterSystems IRIS dynamically adjusts this limit based on system resources, up to a maximum of 16 workers.
5. In the **Default workers** field, select the default number of worker jobs assigned to a work group, when a work group in this category is created and no worker job count is specified. By default, InterSystems IRIS dynamically adjusts this value based on system resources, up to a maximum of 8 workers.
6. In the **Max workers** field, select the maximum number of worker jobs kept in the pool of jobs servicing requests in this category. If the number of jobs requested by a work group in this category is greater than this limit, then InterSystems

IRIS provides workers only up to this maximum value. By default, InterSystems IRIS dynamically adjusts this value based on system resources, up to a maximum of 16 workers.

9.2 Editing or Deleting a Work Queue Manager Category

From the **Work Queue Manager Categories** page, you can modify the maximum number of workers for an existing category and delete user-specified categories as follows:

1. Navigate to **System Administration** > **Configuration** > **System Configuration** > **WQM Categories**.
2. Click **Edit** or **Delete** for the relevant category.

10

Maintaining Local Databases

You can review and maintain local databases on the **Databases** page of the Management Portal (**System Operations** > **Databases**). From this page, you can view the following information:

- Databases General Information — Overview information for all databases.
- Databases Free Space Information — Free space information for all databases.
- Database Details Page — Specific information for individual databases. You can also perform maintenance operations to increase free space from this page.

You also have the option to perform an integrity check using the buttons at the top of the page. For more information, see Verifying Structural Integrity.

Most of the information and operations described in this section can also be found and performed using the ^DATABASE command line utility.

10.1 Databases General Information

The **Databases** page (**System Operation** > **Databases**) contains a list of all local databases. For each local database, you see the following information:

Table 10–1: Local Databases Information

Column Heading	Definition
Name	The database name; click this name to display more details. See Database Details Page.
Directory	The system directory in which the database resides.
Max Size (GB)	The maximum size allocated to which the database can grow, in gigabytes.
Size (MB)	The current allocated size of the database, in megabytes. **Note:** This field measures the logical size of the database. As a result, the size reported here may be lower than the physical size of the database, in particular for the IRISTEMP database.

Column Heading	Definition
Status	The status of the database: mounted (including which permissions it has), unmounted, or dismounted. • A mounted database is one for which **Mount Required at Startup** is selected, as described in Edit Database Properties, and which therefore must be mounted for InterSystems IRIS® data platform to start or become primary in a mirror; in this case, it is always mounted and accessible when InterSystems IRIS starts. Alternatively, it is a previously unmounted database that has been mounted dynamically when you accessed it or explicitly mounted it; in this case, it remains mounted until you explicitly dismount it or restart/stop InterSystems IRIS. • An unmounted database is one for which **Mount Required at Startup** is not selected and which therefore does not need to be mounted for InterSystems IRIS to start or become mirror primary, and has been neither accessed nor explicitly mounted; it is mounted dynamically when you access it or explicitly mount it, and remains mounted until you explicitly dismount it or restart/stop InterSystems IRIS. • A dismounted database is one that has been explicitly dismounted; it is inaccessible until you explicitly mount it or restart/stop InterSystems IRIS (that is, a dismounted database is not mounted dynamically if you try to access it). To permanently dismount a database you must remove it from the configuration, as described in Local Databases.
Encrypted	Indicates whether or not the database is encrypted.
Journal	Indicates whether globals in the database are journaled with a `Y` or an `N`.

In addition, the page contains a filter bar that you can use to control the number of databases displayed. For example, to list only the system databases, you might enter `IRIS*` in the **Filter:** text box; and/or to list only five databases per page, enter `5` in the **Page size:** text box; and/or to limit the number of rows displayed to three, enter `3` in the **Max rows:** text box (a **+** sign displayed with the number in the **Results** field indicates there are additional databases that meet the specified criteria, but they are not displayed).

10.2 Database Free Space Information

Managing the free space (empty blocks) in a database is an important aspect of database maintenance. To see free space information, you can display the Free space view of the Management Portal or use the **^%FREECNT** utility. Remember that the size and free space attributes of a database in normal operation change continuously, and that numbers reported by the Portal or the utility at a given point in time are approximations only.

If you determine that a database has more free space than is necessary, you may compact and truncate the database as described in Database Details Page.

10.2.1 Display Free Space Information Using the Management Portal

To display free space information, which shows information about the amount of free space on each local database, navigate to the **Databases** page (**System Operation** > **Databases**) and click the **Free space view** radio button. The following table describes the information displayed:

Table 10–2: Local Databases Free space Information

Column Heading	Definition
Name	The database name; click this name to display more details. See Database Details Page.
Directory	The system directory in which the primary volume of the database resides.
Max Size	The maximum allocated size to which the database can grow, in gigabytes. The default is unlimited when you create a database.
Size	The current allocated size of the database, in megabytes. **Note:** This field measures the logical size of the database. As a result, the size reported here may be lower than the physical size of the database, in particular for the IRISTEMP database.
Expansion Size	Size (in MB) by which to expand the database. The default and recommended setting is zero (0) when you create a database, which indicates the use of system defaults (12% of the current size or 10 MB, whichever is larger). Under this setting, the expansion size will not be greater than 1GB.
Available	The amount of free space (in MB) available in the database.
% Free	The percentage of free space available in the database.
Disk Free Space	The amount of space free on the volume.
Status	The status of the directory, which indicates if the database is mounted and with what permissions.

For information about performing free space management operations from the Management Portal, see Database Details Page.

10.2.2 Display Free Space Information Using ^%FREECNT

InterSystems IRIS also provides the **^%FREECNT** utility, which you run by entering **do ^%FREECNT** in the Terminal, to display the free space available in a database.

When using **^%FREECNT** in the %SYS namespace, you can choose to display the free space of all databases by entering an asterisk (*) at the prompt, or enter one database directory name. For example:

```
%SYS>do ^%FREECNT

Database directory to show free space for (*=All)? *

Databases Selected
------------------
c:\MyIris\mgr\
c:\MyIris\mgr\irisaudit\
c:\MyIris\mgr\irislib\
c:\MyIris\mgr\irislocaldata\
c:\MyIris\mgr\iristemp\
c:\MyIris\mgr\user\
Device:
Right margin: 80 =>


                            Database Free Space
                              Feb 15 2012  7:25 PM
Database                           Max Size  Size    Available %Free   Disk Free
c:\MyIris\mgr\                    Unlimited 191MB   19MB      9.94    60.79GB
c:\MyIris\mgr\irisaudit\          Unlimited 1MB     0.43MB    43      60.79GB
c:\MyIris\mgr\irislib\            Unlimited 319MB   27MB      8.46    60.79GB
```

```
c:\MyIris\mgr\irislocaldata\      Unlimited 1MB     0.55MB    55      60.79GB
c:\MyIris\mgr\iristemp\           Unlimited 4MB     1.5MB     37.5    60.79GB
c:\MyIris\mgr\user\               Unlimited 1MB     0.43MB    43      60.79GB
```

In a namespace other than %SYS, the utility shows the free space of the databases in that namespace. For example:

```
USER>Do ^%FREECNT


Databases Selected
------------------
c:\MyIris\mgr\user\
Device:
Right margin: 80 =>


                            Database Free Space
                              Feb 15 2012  7:28 PM
Database                           Max Size  Size     Available %Free   Disk Free
c:\MyIris\mgr\user\                 Unlimited 1MB      0.52MB    52      42.72GB
```

Note: A `<-` flag (in the %Free column) indicates that the percentage of free space in the specified database has dropped below 5%. Ensure that there is enough space on the file system to handle database expansion.

The **^DATABASE** utility is another way to display free space information, as well as perform other database tasks. For example, you can compact globals in a database, or you can recreate a database, which lets you clear the data in an existing database without changing the database's name or size. See ^DATABASE for information about the utility.

Note: The data structures used by InterSystems IRIS are self-balancing and suffer no performance degradation over time. It is never necessary to take a database down to rebuild it nor to compress data or indexes to regain performance.

10.3 Database Details Page

The **Database Details** page displays detailed information about any database. To access this information from the Management Portal:

1. Display the **Databases** page (**System Operation** > **Databases**).
2. Click the name of the database for which you would like to see more details.

Along the top of the **Database Details** page is a row of buttons, which allow you to perform various database operations:

- **Mount** / **Dismount** — Mount or dismount a database. The new status remains in effect until you explicitly change it or restart/stop InterSystems IRIS; to permanently dismount a database, you must remove it from the configuration.
- Compact — Move free space distributed throughout a database to its end.
- Truncate — Return free space at the end of a database to the underlying file system.
- Defragment — Relocate global blocks so that the blocks representing a given global are in continuous sequence.

The page also includes information specific to the selected database, which is organized into a General Information table and a Database Size table:

Table 10–3: General Information

Field	Definition
Directory	Name of directory where this database resides.

Field	Definition
Resource Name	Resource name assigned to this database.
Mounted	Indicates whether this database is mounted.
Read Only	Indicates whether this database is mounted as read-only.
Read Only Reason	The reason the database is mounted as read-only.
Encrypted	Indicates whether this database is encrypted.
Encryption Key ID	Indicates encryption key ID.

Table 10–4: Database Size

Field	Definition
Block Size	Block size (bytes) for this database.
Blocks	Current number of blocks within this database.
Max Size	Maximum size (MB) allowed for this database.
Size	Current size (MB) of this database.
Expansion Size:	Amount (MB) this database will expand by.
Available Space	Available space within this database.
% Free Space	% Free space within this database.
Disk Free Space	Free space on disk containing this database.
Last Expansion Time	Last time this database expanded.
Full	Indicates that this database is full.

10.4 Compacting a Database

Compacting a database moves free space distributed throughout the database to its end by relocating global blocks. You can then return the free space to the underlying file system by truncating the database. (You can also compact globals; see Compact Globals in a Database.)

When you *compact* a database, you specify the amount of the available free space to be positioned at its end, and the operation rearranges global blocks to ensure that at least that amount of free space is located at the end. (The operation cannot create more free space, so it can never place more at the end than the total available amount.)

For example, suppose the size of a database is 50 MB, with 15 MB of that being free space, and 5 MB of that free space already positioned at the end of the database. If you compact the database and specify more than 5 MB but less than 15MB, global blocks are moved from the end of the database to the beginning until the free space at the end equals the amount you specified; if you specify 15 MB, all possible global blocks are moved to the beginning.

To compact a database:

1. Navigate to the **Databases** page (**System Operations** > **Databases**).
2. Click the name of the database you want to compact. This takes you to its Database Details page.

3. Click **Compact** on the ribbon at the top of the page. This displays the **Compact Database** dialog box, which shows the name and location of the database, its current size, the total available free space, and the amount of free space currently at the end of the file.
4. Specify the amount of free space you want at the end of the file in the **Target free space (in MB) at end of file** text box. Your entry must be within the stated range. Once you have entered an amount, click **OK**. If all of the free space is already at the end, or there is no free space, the prompt does not appear and the **OK** button is disabled.

   Note: For a number of reasons, the operation may move more free space than the amount you specify. Conversely, because the numbers reported are approximations, it is possible that not all of the free space displayed can actually be moved.

5. When the task is complete, a **Background Task Info** dialog box appears. Optionally, you can also view the task on the **Background Tasks** page (**System Operation** > **Background Tasks**).
6. Click **Close** to redisplay the **Database Details** page. The page should automatically refresh to show the new database information.

Note: The compact database operation is designed to run concurrently with normal database activity. The operation does consume some system resources, however, and may not complete if the system is under extremely high load. For these reasons, InterSystems recommends running this and other database reorganization operations (including compacting and defragmenting globals) during off-peak hours, and running only one such operation on a system at a time.

10.5 Truncating a Database

Truncating a database returns free space from the end of the database to the underlying file system. A database is often truncated after being compacted, which moves free space to the end of the database.

When you *truncate* a database, you specify a target size for the database. If there is sufficient free space at the end of database, the operation removes enough to reduce the database to the target size; if not, it removes all that can be removed. (To find out how much of a database’s free space is positioned at the end, compact the database; you do not need to complete the operation to display current total available free space and the amount at the end.)

Note: This feature is not applicable to databases with raw volumes.

To truncate a database:

1. Navigate to the **Databases** page (**System Operations** > **Databases**).
2. Click the name of the database you want to truncate. This takes you to its Database Details page.
3. Click **Truncate** on the ribbon at the top of the page. This displays the **Truncate Database** dialog box, which shows the name, location, and current size in megabytes (MB) of the selected database.
4. Enter the **Target File Size (MB)**, which must be less than the current size, and click **OK**. Enter `0` to remove all possible free space from the end of the file.

   Note: Free space for truncation purposes is different than free space for storing data. Certain control structures can be deleted during truncation if they are not being used for data storage. This can lead to a difference in reported free space.

10.6 Defragmenting a Database

Defragmenting a database rearranges global blocks within the database so that all of the blocks containing data for a given global are in consecutive sequence. The operation does not place big string blocks or pointer blocks from a global in sequence, but it does locate them in a contiguous area. As part of the process, the **Defragment a database** option compacts all globals in the same manner as the **Compact globals in a database** option, but with a target density of 70%. (If this is lower than the current global block density of the database, the size of the database does not increase.)

Note: The IRISTEMP database cannot be defragmented.

In general, it is not necessary to run defragmentation on any regular basis. Some workloads, however, particularly those that read large portions of a database sequentially, can benefit from having global blocks organized sequentially.

The defragmentation process requires a certain amount of free space at the end of the database. For this reason, the following possibilities exist:

- If there is enough free space at the end to perform the operation, it completes without any changes to the database beyond global defragmentation.
- If there is not enough free space in the database, the database is expanded as necessary. When defragmentation is complete, you can truncate the database to remove the added free space.
- If there is not enough free space in the database but there is significant free space that could be moved to the end, you are informed of this. In this case, compacting the database before you choose the **Defragment a database** option reduces the amount of expansion required to complete the defragmentation operation.

To defragment the globals in a database, use the following procedure:

1. Navigate to the **Databases** page (**System Operations** > **Databases**).
2. Click the name of the database you want to defragment. This takes you to its Database Details page.
3. Click **Defragment** on the ribbon at the top of the page. This displays the **Defragment Database** dialog box, which shows the name, location, and current size in megabytes (MB) of the selected database, as well as a message describing the free space needed to defragment.
4. Click **OK**.
5. When the task is complete, a **Background Task Info** dialog box appears. Optionally, you can also view the task on the **Background Tasks** page (**System Operation** > **Background Tasks**).
6. Click **Close** to redisplay the **Database Details** page. The page should automatically refresh to show the new database information.

Important: The defragment operation temporarily relocates all of the data in the database, regardless of the degree of global fragmentation in the database prior to running the operation. Subsequent runs of the operation consume similar amounts of resources but do not provide any additional benefit.

Note: The defragment operation is designed to run concurrently with normal database activity. The operation does consume some system resources, however, and may not complete if the system is under extremely high load. For these reasons, InterSystems recommends running this and other database reorganization operations (including compacting a database and compacting globals) during off-peak hours, and running only one such operation on a system at a time.

Global defragmentation can involve a temporary increase in the size of the database being compacted. If this causes the database to reach its configured maximum size (see Local Databases), or if expansion is not possible because there is insufficient space available on the storage volume, the operation is canceled.

10.7 Compacting Globals in a Database

Another way to manage database space is to compact the globals in a database using the **^DATABASE** routine. Compacting globals consolidates global data into fewer blocks, increasing the amount of free space in a database.

When globals are created and updated, InterSystems IRIS typically allocates data in a manner that fills global blocks to about 70% of capacity. (Globals that have grown entirely in collation order may be allocated at closer to 90%.) In general, allowing InterSystems IRIS to manage global block density automatically is sufficient. However, some nonsequential patterns of data deletion may reduce average global block density considerably.

Note: To see the current density of the global blocks in a database on a global by global basis, you can run an integrity check (as described in Verifying Structural Integrity) and examine the `Data Level` output for each global.

When you compact globals, you specify a desired global block density (90% by default) and the operation attempts to come as close to this as possible by consolidating data—for example, rearranging global data that is spread across three blocks into two. Typically (but not always), compacting globals yields a meaningful increase in available free space within a database. (If you specify a target density that is lower than the current global block density of the database, the size of the database does *not* increase.)

To compact the globals in a database, use the following procedure:

1. Open the Terminal and change to the **`%SYS`** namespace.
2. Enter `do ^DATABASE`, and select `7) Compact globals in a database` from the menu.
3. Specify the directory of the database on which you want to run the operation. You can specify multiple databases by entering `?` at the `Database directories to compact?` prompt and then entering a list of numbers.
4. Indicate that you want to compact all globals, or instead enter a list of individual globals to be compacted.
5. Specify the target average global block density, respond to the rest of the prompts, and confirm.

Note: The compact globals operation is designed to run concurrently with normal database activity. The operation does consume some system resources, however, and may not complete if the system is under extremely high load. For these reasons, InterSystems recommends running this and other database reorganization operations (including compacting a database and defragmenting globals) during off-peak hours, and running only one such operation on a system at a time.

Global compaction can involve a temporary increase in the size of the database being compacted. If this causes the database to reach its configured maximum size (see Local Databases), or if expansion is not possible because there is insufficient space available on the storage volume, the operation is canceled.

11

Controlling InterSystems IRIS Processes

An InterSystems IRIS system runs a number of processes. Application code as well as InterSystems IRIS system code executes within these processes. There are three categories of InterSystems IRIS processes:

- User processes, created when a user connects to InterSystems IRIS.
- Background processes, created when a user issues an ObjectScript **Job** command, or by the Management Portal or a utility (see Using the Background Tasks Page).
- InterSystems IRIS system processes.

In this page, the word *process* by itself refers to both user and background processes.

11.1 Available Options

You can manage and control processes using the Management Portal as follows:

Table 11–1: Process Management Functions

Function	How to Access Function From The Portal
Display process information	Display the **Processes** page (**System Operation** > **Processes**).
Display process details	Display the **Processes** page, then click **Details** in the right hand column of the selected process to display the **Process Details** page.
Suspend/resume a process	Display the **Processes** page, then click **Details** in the right hand column of the selected process to display **Process Details** page. Then click **Suspend** or **Resume** on the operations bar, as desired.
Terminate a process	Display the **Processes** page, then click **Details** in the right hand column of the selected process to display the **Process Details** page. Then click **Terminate** or **Terminate with <RESJOB> Error** on the operations bar, as desired.
Display process variables	Display the **Processes** page, then click **Details** in the right hand column of the selected process to display the **Process Details** page. Then click the **Variables** tab to display the process variables.
Broadcast messages to terminals	Display the **Processes** page and click the **Broadcast** button to open the **Broadcast** dialog.

11.2 Display Process Information

To display all the active processes on the system and basic information about each, navigate to the **Processes** page (**System Operation** > **Processes**), which displays a table of the processes with statistics about each in columns.

The following table describes the process information available for display:

Table 11–2: Process Column Information

Column Heading	Definition
Job #	Index of the Processes table.
Process ID	Operating system process identification number (PID).*
Total CPU Time in ms	Total amount of system and user CPU time, in milliseconds, that the process took to execute.
User	Name of the user who owns the process.
Device	Current device the process is using. This can be: • `|TCP|`*IP_address*`:`*Port_number* — For an outbound connection from the instance. • `|TCP|`*Port_number* — For the superserver. • `//./nul` — The null device. This discards any output; if you attempt read from it, there will be no data.
Namespace	Namespace in which the process is running.

Column Heading	Definition
Routine	Name of the routine that the process is currently executing.
Commands	Number of commands executed.
Globals	Number of global references, including updates, executed (database reads and writes) since the process entered InterSystems IRIS.
State	Process state. See the *State* property of the %SYS.ProcessQuery class for an explanation of each state.
Client Name	Name of the client system that connected to, or initiated the connection to, the process.
Client EXE	Name of the executable that called the process.
Client IP	IP Address of the system that initiated the process.
O/S Username	Username assigned to the process by the operating system.
Details	Button appears if you have authority to maintain this process. See Display Process Details.

* An asterisk (*) appears next to the process id if the user entered InterSystems IRIS in programmer mode. A plus or minus sign appears next to Callin processes:

- + Process is in InterSystems IRIS
- – Process is not in InterSystems IRIS

The Callin API is an InterSystems IRIS facility that lets you execute and evaluate ObjectScript commands and expressions from within C programs.

11.3 Display Process Details

The **Process Details** page displays detailed information about any process. To access this information from the Management Portal:

1. Display the **Processes** page (**System Operation** > **Processes**).
2. Click **Details** in the row of the appropriate process. (This option exists only on processes that you have authority to maintain.)
3. Optionally select the **SQL table & statement info** check box. This option adds SQL-related information to the display.

The page also includes information specific to the selected process, which is organized into a General Information table, a Client Application Details table, and a Execution Details table:

Table 11–3: General Information

Field	Definition
Process ID	Process ID (PID) number of this process.
User Name	Name of the user currently logged in for this process.
Login Roles	Login roles for the process.

Field	Definition
Escalated Roles	Additional roles granted to the process. These roles plus the Login Roles is the total set of roles granted to the process.
OS User Name	Username assigned to the process by the operating system.
NameSpace	Namespace in which the process is executing.
Process Priority	Priority level of this process.
Global References	Number of global references made by this process.
Private Global References	Number of private global references made by this process.
Commands Executed	Number of commands executed by this process.
Memory Limit	Amount of memory (KB) allocated for use by this process.
Memory Peak	Peak amount of memory (KB) used by this process.
Memory Used	Amount of memory (KB) currently in use by this process.
Total CPU Time	Total amount of system and user CPU time, in milliseconds, that the process took to execute.
Private Global Blocks	Number of private global data blocks used by this process.
Current Device	Name of the I/O device currently in use by this process.
Open Devices	List of devices currently opened by this process.
Lock	Lock information for this process. Click the link at top of the detail box for additional details (mode, counts, and full reference).

Table 11–4: Client Application Details

Field	Definition
Client Name	Node name of the client that is connected, or initiated the connection, to this process (if any).
EXE Name	Name of the executable client application client connected to this process (if any).
Client IP Address	IP address of the executable client application client connected to this process (if any).
Info	User-defined information (if any).

Table 11–5: Execution Details

Field	Definition
Process State	Current execution state of this process.
In Transaction	Indicates whether or not this process is currently within a transaction.
Last Global Reference	Last global referenced by this process.
Last SQL Table Reference	Last SQL table referenced by this process, if any. (Select the **SQL table & statement info** check box in order to see this data.)
Routine	Name of the routine this process is currently executing.
Source Location	Last reported source location (routine name plus offset) of this process.

Field	Definition
Source Line	Last reported line of source code executed by this process, if available.

Stopping a Process

From this page you can also stop or resume a process. You can stop a process in one of the following ways:

- Suspend a process with the intention of resuming the process later.
- Terminate a process, which entirely cancels the process.

11.3.1 Suspend or Resume a Process

You may want to suspend a process if you are not sure what it is doing and want to investigate, or if a more important process is trying to run and needs the CPU cycles. To access this option from the Management Portal:

1. Display the **Processes** page (**System Operation** > **Processes**).
2. Click **Details** in the row of the appropriate process. This option only exists on processes that you have authority to maintain.
3. Click **Suspend** on the options bar.

You may resume a suspended process at any time by clicking **Resume** from the same page.

11.3.2 Terminate a Process

You may want to terminate a process if it becomes unresponsive or is affecting other processes or users. To access this option from the Management Portal:

1. Display the **Processes** page (**System Operation** > **Processes**).
2. Click **Details** in the row of the appropriate process. (This option exists only on processes that you have authority to maintain. The portal displays the **Process Details** page for the process you selected.
3. Click **Terminate** on the options bar.

   Optionally, to log the status of the process when it terminates, select the **Terminate with RESJOB Error** check box.

   Note: This option is enabled by default.

4. Click **Yes** to confirm that you want to terminate the process. There is no way to resume a terminated process.

11.3.3 Display Process Variables

The **Process Variables** page displays all the variables used in the selected process giving the global name and the value of the global. To access this information from the Management Portal:

1. Display the **Processes** page (**System Operation** > **Processes**) page.
2. Click **Details** in the row of the appropriate process. (This option exists only on processes that you have authority to maintain.) The portal displays the **Process Details** page for the process you selected.
3. Click **Variables** on the options bar.

If you have selected the **SQL table & statement info** check box, this table includes a column via which you can directly view the cached squery, if any.

11.4 Broadcast Messages to Terminals

You can broadcast messages to the terminals associated with a selected process or all processes; this utility is useful, for example, to ask people to sign off the system. However, you must use it carefully or you may cause messages to appear in the middle of reports that may be printing at the time.

The utility temporarily takes control of each terminal as it sends the message. Once the terminal receives the message, the previous process continues. The message appears on the terminal screen; it may disrupt the screen display, but it does not affect user input. The message does not appear in windows running ObjectScript utilities.

To broadcast a message to the terminals associated with a selected process, do the following in the Management Portal:

1. Display the **Processes** page (**System Operation** > **Processes**).
2. Click **Broadcast** (on the options bar) to open the **Broadcast** window.
3. Enter the message to broadcast in the text box.

   (The dialog box notifies you if there are no active processes that can accept a message; you do not see a message text box or list of processes. Click **Close**.)
4. Select the appropriate check boxes for the appropriate processes (PIDs) to receive the broadcast message. Use the **Select All** and **Clear All** buttons accordingly to help with the selection.
5. Click **Broadcast**.
6. After the completed message displays, click **Close**.

12

Using the Task Manager

To access the Task Manager, navigate to **System Operation** > **Task Manager**.

You can also interact with the Task Manager using the Terminal, as described in Using ^TASKMGR. See Tasks for other methods of managing tasks.

Note: The Task Manager polls every 60 seconds to see if there are any Tasks to be run. When you click **Perform Action Now** to schedule a Task, there may be a delay of up to 60 seconds before the newly scheduled Task actually runs.

12.1 New Task

The **New Task** option starts the **Task Scheduler Wizard**. This tool allows you to schedule a new task to run.

Note: You can also use the **Task Scheduler Wizard** to edit an existing task by selecting the existing task from the Task Schedule page.

1. On the first page of the **Task Scheduler Wizard**, provide the following details about the new task:
    - **Task name**
    - **Description**
    - **Namespace to run task in** — Choose from the list of defined namespaces in which to run the task
    - **Task type** — Choose from among the listed tasks. For details about the available tasks, see Task Types below.

      Note: Depending on the task type selected, you may be presented with a form in which to specify additional information; for example, if you are scheduling an `IntegrityCheck`, the form prompts you for **Directory**, **Filename**, and **KeepDays** (number of days to keep the file).

2. **Task priority** — Choose from **Priority Normal**, **Priority Low**, or **Priority High**. For more information about priority, see Priority.
3. **Run task as this user** — Choose from the list of defined users. To choose a different user than the one you are logged in as, you must have the **`%Admin_Secure:Use`** privilege.

> **Note:** If the chosen user is disabled, the task is suspended until the user is enabled and the task is resumed manually. This does not apply to built-in system tasks, which run even when the chosen user is disabled.
>
> Each task consumes a license unit, with the license user ID based on the InterSystems IRIS® data platform username you select; see License Login Special Considerations for more information.

4. **Open output file when task is running** and **Output file** — If the task creates output, the log file is written to the directory specified.
5. **Suspend task on error?** — Specify whether the task will be rescheduled and continue to run after an error, or suspended. The default is **No**, to reschedule and run the task after an error.
6. **Reschedule task after system restart?** — Specify whether you want to reschedule the task when the system restarts (that is, if the system is down when the task is scheduled to run):
    - **No** specifies that the task should run when the system restarts.
    - **Yes** specifies that the task should be rescheduled for the next logical time after the system restarts.
7. **Send completion email notification to** — If you have configured email settings for the Task Manager (see Configuring Task Manager Email Settings), enter a comma-separated list of email addresses to which a notification should be sent when the task ends successfully.
8. **Send error email notification to** — If you have configured email settings for the Task Manager, enter a comma-separated list of email addresses to which a notification should be sent when the task ends in error.
9. **How should task run for Mirror** — If this instance is a mirror member, specify the type of member the task can be run on:
    - primary failover member only
    - backup failover member and async members only (all except primary)
    - all mirror members (primary, backup, and asyncs)

    > **Important:** For a task to run on a mirror member, this option must be set for all tasks on the mirror member. Adding an instance to a mirror member does not automatically update this option in user-defined tasks. As such, you must either:
    >
    > - Define this setting when you create a task even if the instance is not a mirror, so it can run if the instance is added to a mirror.
    > - Make sure you review all user-defined tasks when an instance is added to a mirror and set **How should task run for mirror**.

    > **Note:** When the status of a failover member is in transition, for example when the backup is in the process of taking over as primary, the Task Manager does not run any tasks on that member until its status as primary or backup is established.

    After you have filled in all the necessary fields, click **Next**.
10. On the second page of the **Task Scheduler Wizard**, specify when the new task should run. The options are:
    - **Daily** — A daily interval (for example, every other day, or every third day).
    - **Weekly** — A weekly interval on specific days of the week.
    - **Monthly** — A monthly interval on a specific day of the month.

        > **Note:** If the chosen date does not exist for a particular month, the closest existing date in that month is used.

- **Monthly (by day)** — A monthly interval on a specified day of the week (for example, first Monday, or third Wednesday).
- **After another task completes** — After a specified task runs.
- **On demand** — Only when manually executed.

The **Daily**, **Weekly**, **Monthly**, and **Monthly (by day)** fields allow you to specify what time during the day the task should run:

- **Start Date** — The first day the scheduled task should run.
- **End Date** — The last day the scheduled task should run.
- **Task execution details**, which specify what happens when a task executes as follows:
  - **Run once at this time** — At the specified time, the task executes once.
  - **Run every** ____ — During the specified time range, the task executes repeatedly at the specified interval.

Note: By default, all new tasks expire if they miss their scheduled time for any reason. If you wish to change this behavior, see Changing Task Expiration Behavior below.

After you have filled in all the necessary fields, click **Finish** to schedule the task.

12.2 Task Types

This section describes the predefined task types you can choose when creating or editing a task. Some tasks are only available from certain namespaces.

InterSystems IRIS includes a number of tasks that run by default. This table notes these defaults, which you may edit from the Task Schedule page.

Note: You can define your own task types using the %SYS.Task.Definition API; for details, see the class documentation in the *InterSystems Class Reference*.

Task Type	Corresponding Default Task	Description
CheckLogging	**Check Logging activity** — runs nightly; sends an alert after two days.	Checks that logging mechanisms (such as ^ISCSOAP) have not been left running unintentionally. Creates an alert after *DaysForAlert* days, and shuts off logging after *DaysForShutoff* days.
CumuIncrDBList	None	Runs a cumulative backup of databases in the defined list.
DiagnosticReport	**Diagnostic Report** — runs on demand.	Collects and delivers diagnostic reports to the WRC.
FullAllDatabases	None	Runs a full backup of all databases.
FullDBList	None	Runs a full backup of databases in the defined list.

Task Type	Corresponding Default Task	Description
IncrementalDBList	None	Runs an incremental backup of databases in the defined list.
IntegrityCheck	**Integrity Check** — disabled by default.	Runs an integrity check of databases in the namespace. The Integrity Check task only appears when the %SYS namespace is selected.
InventoryScan	**Inventory Scan** — runs on demand.	Compiles an inventory scan of the instance.
PurgeAudit	**Purge Audit Database** — runs after the **Switch Journal** task.	Purges the audit database after a specified time following a journal switch.
PurgeBackupLog	**Purge Backup Log** — runs nightly.	Purges the backup log after a specified time following a database backup.
PurgeErrorsAndLogs	**Purge errors and log files** — runs nightly.	Purges error globals and messages.log based on settings in the iris.cpf file (see ErrorPurge and MaxConsoleLogSize parameters in the [Startup] section of the *Configuration Parameter File Reference*). **Note:** When System Monitor is running, it monitors and limits the size of the messages log.
PurgeJournal	**Purge Journal** — runs nightly.	Purges journal files that meet the purging criteria.
PurgeTaskHistory	**Purge Tasks** — runs nightly.	Purges Task Manager history files.

Task Type	Corresponding Default Task	Description
RunLegacyTask	None	Runs a legacy task (i.e. your own programmatic task). The line you enter in the text box (for example, **do ^MyCode**) must be executable in the Terminal. **Note:** Do not use the local variable *Status* in `RunLegacyTask` code. InterSystems IRIS uses this variable, and if it is modified by `RunLegacyTask` code, the task is likely to end with an error status. A legacy task that fails to complete due to an error is marked **Suspended due to error** and is not scheduled to run until the error is corrected and you resume the task.
SecurityScan	**Security Scan** — runs nightly.	Disables expired user accounts and expires user passwords and web session tokens.
SwitchJournal	**Switch Journal** — runs nightly.	Switches to a new journal file.

12.3 On-demand Task

The **On-demand Task** page (**System Operation** > **Task Manager** > **On-demand Task**) lists the tasks you have scheduled as *on-demand*. The list includes the task name, a description, and an option to **Run** the task from this page. You can sort the information in the table by clicking any column heading. When you click **Run**, the **Run Task Wizard** page displays the task name and ID, and the date and time the task will run; click **Perform Action Now** to confirm the information and schedule the task.

12.4 Upcoming Tasks

The **Upcoming Tasks** page (**System Operation** > **Task Manager** > **Upcoming Tasks**) lists the tasks scheduled to run within a certain interval. To select an interval, click an option in the **Scheduled to run:** search pane to the left of the task list. If you select the **To a date** option, you can either enter a date in `yyyy-mm-dd` format or click the calendar icon to select a date from the calendar.

You can sort the information in the task list by clicking any column heading. You can **Suspend** or **Resume** the scheduling of each task by clicking the appropriate option:

- **Suspend** — Lets you suspend the task; a **Do you want to reschedule task when task is supposed to run?** drop-down list lets you specify:
  - **No.** Upon resuming the task, the Task Manager attempts to run missed instances of the task before returning to the normal schedule.
  - **Yes.** Upon resuming the task, the Task Manager returns to the normal schedule without attempting to run missed instances.
- **Resume** — Lets you resume a suspended task.

12.5 Task Schedule

The **Task Schedule** page (**System Operation** > **Task Manager** > **Task Schedule**) lists all scheduled tasks. You can sort the information in the table by clicking any column heading. You can view **Details** or **History**, as well as **Run**, a scheduled task by clicking the appropriate option:

- **Task Name** — Lets you view details about the task and perform operations on it.
- **History** — Displays the task's history.
- **Run** — Lets you run the task. A **Run Task** wizard displays the task name and ID, and the date and time the task will run; click **Perform Action Now** to confirm the information and schedule the task.

12.5.1 Task Details

To display detailed information about a scheduled task and perform one of several operations on it, click the task's name in the **Task Name** column. The **Task Details** page displays information and execution details about the selected task. You can perform one of the following operations on the task by clicking the appropriate button:

- **Edit** — Change the task definition and schedule using the **Task Scheduler Wizard**.
- **History** — View the task's history.
- **Resume**/**Suspend** — Suspend a task/resume a suspended task, as on the Upcoming Tasks page.
- **Delete** — Delete the task permanently.

  Note: You cannot delete a system task.
- **Export** — Export a task to a file that can later be imported, including by another InterSystems IRIS instance.
- **Run** — Schedule the task. When you click **Run**, the **Run Task Wizard** page displays the task name and ID, and the date and time the task will run; click **Perform Action Now** to confirm the information and schedule the task.

Note: Some of the actions described are unavailable while a task is running.

12.5.2 Scheduled Task History

To display history information about an individual scheduled task, click the **History** link in the row of the item. The **Task History** page displays detailed history for the selected task. The **Result** column indicates the outcome the last time the task was run, showing either **Success** or an error message. You can sort the information in the table by clicking any column heading.

The **Details** link at the top of the page displays the **Task Details** page for the selected task.

12.6 Task History

The **Task History** page (**System Operation** > **Task Manager** > **Task History**) lists the history of all tasks executed by the Task Manager.

You can sort the information in the table by clicking any column heading. To filter out system tasks, select the **Show only user-defined task types** check box at the top of the table.

12.7 Import Tasks

The **Import Tasks** page (**System Operation** > **Task Manager** > **Import Tasks**) lets you import and run a task by browsing to a previously-exported task file, then clicking **Perform Action Now**. For information about exporting tasks to a file, see Task Details in this section.

Note: Tasks can only be imported from, or exported to, instances running the same version of InterSystems IRIS.

12.8 Using ^TASKMGR

The **^TASKMGR** routine allows you to configure the Task Manager using the Terminal. Except when noted otherwise, **^TASKMGR** and the Management Portal contain the same options for configuring tasks; the one you use is a matter of preference.

To use **^TASKMGR**:

1. Open the Terminal.
2. Enter `set $namespace = "%SYS"` to change to the %SYS namespace.
3. Enter `do ^TASKMGR`.

For more details about how to schedule or edit a task programmatically, see the %SYS.Task class documentation in the *InterSystems Class Reference*.

12.8.1 Changing Task Expiration Behavior

By default, all tasks are configured to expire if they miss their scheduled time. This could happen for a number of reasons; for example, when InterSystems IRIS is down during the scheduled time, or if the previous run of the task extends beyond the next scheduled time. When a task expires, it does not run until the next scheduled time.

The **^TASKMGR** routine (but not the Management Portal) contains the option to change this behavior, such that a scheduled run is never skipped. To do so:

1. Open the Terminal.
2. Enter `set $namespace = "%SYS"` to change to the %SYS namespace.
3. Enter `do ^TASKMGR`.

4. Select option `2.`
5. Enter the task number you want to edit.
6. Press Enter to save the defaults of each option until you reach `Task Expires?`
7. Enter `No`.

Alternatively, you can specify a grace period, such that the task does not expire immediately after it is missed. Instead, the Task Manager runs the missed task as soon as it is able. To adjust this grace period of when a specific task expires:

1. Open the Terminal.
2. Enter `set $namespace = "%SYS"` to change to the %SYS namespace.
3. Enter `do ^TASKMGR`.
4. Select option `2.`
5. Enter the task number you want to edit.
6. Press Enter to save the defaults of each option until you reach `Task Expires?`
7. Enter `Yes`.
8. At the `Expires in how many days?` prompt, enter the number of days before the task should expire.
9. At the `Expires in how many hours?` prompt, enter the number of hours before the task should expire.
10. At the `Expires in how many minutes?` prompt, enter the number of minutes before the task should expire.

12.9 Custom Tasks

To define custom tasks, define a subclass of %SYS.Task.Definition. In this subclass, implement the **OnTask()** method which should perform the required work and return a %Status value.

You can require a specific privilege to execute this task. To do so, specify the *RESOURCE* class parameter in this class. Specify the value in the form *resource:permission*. For example:

Class Member

```
Parameter RESOURCE As String = "%Admin_Manage:USE";
```

For additional information, see the class reference.

After you compile this class, it is available to be added to the Task Manager schedule as described on this page. It is listed there as a User task (as opposed to a System task).

13

Using the Background Tasks Page

A background task is an asynchronous job process that runs in the background, independently of the process that created it. A background task is created when a user issues an ObjectScript JOB command, or by the Management Portal or a utility to execute a job without requiring the user to wait for completion. For example, when you use the Portal to truncate a database, as described in Truncating a Database, a background task is started.

The **Background Tasks** page (**System Operation** > **Background Tasks**) lists past and active background tasks. You can purge the log of past background tasks at any time.

When a background task is active, the process can also be seen on the **Processes** page (**System Operation** > **Processes**), as described in Controlling InterSystems IRIS Processes.

14

Managing InterSystems IRIS Licensing

This topic contains an overview of the InterSystems IRIS® data platform license system and describes how to manage and monitor licenses.

Important: *InterSystems Terms and Conditions* govern how you may use the licensed InterSystems IRIS software. Occasionally, the implementation may be more lenient. Verify that any license-related code you write conforms to these terms and conditions.

If you use license servers, there are important network considerations to take into account.

14.1 Introduction to License Management

Each InterSystems IRIS instance maintains an independent local view of its license capacity and current use, and each instance requires access to a license key (except evaluation installations). To manage these licenses, there are several possibilities, depending on the number of instances you have:

- You can install and activate a local license key file on each instance.
- You can configure a *license server* to manage key files stored in a central location, which it can then distribute to other instances. In this case, each instance must be configured with the LicenseID of the key file, so that it can retrieve a copy of the key at startup.

  The license server coordinates the views of license use maintained locally in every instance. The license server is not an InterSystems IRIS process; it is unaffected if an InterSystems IRIS instance shuts down. One license server can handle multiple instances. Therefore, you need at most one per host regardless of how many InterSystems IRIS instances run on a host. However, each InterSystems IRIS instance must have a local copy of the authorizing license key file installed. Also, because a license server is started by a running instance, it should be configured to run on systems where you expect an InterSystems IRIS instance to be running consistently.

  License servers can run on any computer where you run an InterSystems IRIS instance. A monitor process (the *license monitor*) sends update messages to the license server, which coordinates license allocation when more than one instance shares a license.
- You can configure more than one license server (typically on different hosts) to provide redundancy. The license software selects one of the license server to be the active server. The other servers are available to take over should the active server fail. When configuring license server, decide which machines to use to host the license server. You can configure it to run on as many hosts as you want, but more than three is excessive.

Multiserver licenses can be shared among cooperating instances, either on the same machine or on different machines. Sharing is permitted only with multiserver keys. To use your multiserver licenses, you must configure one or more InterSystems IRIS license servers to allocate the InterSystems IRIS license units authorized by the key. All instances sharing a key must be configured to use the same license server or set of license servers.

Multiple instances with different license keys and running on different platforms *can* use the same license server to coordinate licensing *as long as* each instance has its own copy of the proper iris.key file *and* all instances authorized by the same key use the same license servers. However license units are not summed across license keys. InterSystems IRIS instances using different license keys do not share license units, and users logged in to two instances using different license keys consume a separate license unit from each key.

14.2 License Servers and Network Considerations

If you use license servers, bear in mind the following points:

- Each license monitor queries a license server via a TCP port (using %SYSTEM.License). Thus it is necessary to make sure that TCP traffic to the license server is permitted.

  Important: If a remote license server is protected by a firewall, the license server port must be open for UDP traffic.

- As part of the built-in redundancy for the license server, the License Monitor on an IRIS Instance checks periodically to make sure it's connected to the active license server. If it finds an IP address different than the one it's currently connected to, it assumes the license server has changed and enters a restart mode which resends all current license usage, resulting in warning messages. To avoid this problem, make sure to specify the location of the license server as a stable, consistent IP address.

14.3 Configuring a License Server

You can add or delete license servers from the **License Servers** page (**System Administration** > **Licensing** > **License Servers**) in the Management Portal. The page displays a list of license servers configured for this instance. When there are multiple license servers configured for this instance, the row of the active license server is shaded.

Note: You can also determine which license server is active using the **$System.License.ShowServer()** method.

Click on the name of a listed license server to update its information, or click **Delete** to remove it. To add a new license server:

1. Click **Create License Server** to configure a license server. The **Create New License Server** box appears on the right edge of the screen.
2. Enter a name for the license server in the **Name** box. The license server name identifies the license server in the configuration and must be unique to a configuration.
3. Enter the IP address to host the license server in the **Hostname/IP Address** box. You can enter the IP address in dotted decimal format (`192.29.233.19`) or in alphabetic format (*mycomputer.myorg.com*). If IPv6 is enabled, you can enter a colon separated format IPv6 address (`2001:fecd:ba23:cd1f:dcb1:1010:9234:4085`).
4. Enter the port number used by the license server in the **Port** box. The license server port number must be a number between `1024` and `65535`; InterSystems IRIS uses a default port number of `4002`. Redundant license servers running

on different hosts do not need to use unique port numbers, but must use port numbers that are not already in use at that IP address.

5. (Optional) Enter the central directory where license keys are stored in the **KeyDirectory** box. For more information, see Loading Keys to the License Server.
6. Click **Save** to create the license server.

After adding or deleting a license server, you must restart the InterSystems IRIS instance.

Note: If separate instances all configure the same license server address and port, they all use the same license server; when this is the case, you should delete the default **LOCAL** license server (127.0.0.1) on each instance. If the same key is loaded on each instance, they share the key; if different keys are loaded on each instance, the license server serves each set of instances using each key separately.

14.4 Loading Keys to a License Server

In addition to managing the capacity of shared licenses, the license server may also distribute keys stored in a central directory to InterSystems IRIS instances.

A KeyDirectory property is defined as part of Config.LicenseServers. If this is filled in, the instance which starts the license server reads any valid *.key files found there at startup and sends them to the license server. Each key file must have a unique LicenseID property. The instance logs a count of files successfully loaded and any errors. You may manually reload key files from the directory to update any licenses by calling **ReloadKeys^%SYS.LICENSE** from the %SYS namespace.

Note: Loading a new key with the same LicenseID as an existing key in the license server key tables marks the existing key as replaced. Requests from instances for that LicenseID receive the most recently loaded key. You can use the existing **$System.License.DumpKeys()** method to view the current state of keys in the license server.

14.5 Activating a License Key

InterSystems IRIS uses license keys to ensure proper operation of its registered sites, to define capacity available and to control access to InterSystems IRIS features. (License keys are not required for evaluation installations.) A license key is provided in the form of a license key file, typically named iris.key.

After installing InterSystems IRIS, use the following procedure to activate your license key. You can always use the same procedure to activate a new license key (that is, upgrade the key) for any installed instance. You can activate a license key placed in any location accessible to the Management Portal; as part of activation, the license key is copied to the instance's *install-dir*/mgr directory as iris.key, if it is not named that already.

Note: On Windows, you can also select a license key during installation (see Installing InterSystems IRIS on Microsoft Windows). When you do this, the license is automatically activated and the license key is copied to the instance's *install-dir*/mgr directory as iris.key; the activation procedure described here is not required.

This section also discusses license troubleshooting and upgrading a license from the operating system command line when all license units are in use.

To activate a license key, use the following procedure:

1. Navigate to the **License Key** page (**System Administration** > **Licensing** > **License Key**). Information about the current active license key is displayed. If no license has yet been activated, this is indicated, for example by the notation **Customer Name: License missing or unreadable**. This page includes a **Print** button to let you easily print the displayed information.
2. Click **Activate License Key** and browse for the license key file you want to activate. When you select a file, information about it is displayed so you can verify that you have the right license key before activating it; for example, that it provides the desired capacity, and has the right expiration date. If the key is not valid, this is indicated in an error message. If a license is currently active, information about the current and selected licenses is displayed side by side. If a restart of the instance after activation is required for the license key to take effect, this is noted and the reason for it is provided. This dialog includes a **Print** button to let you easily print information about both the current active license and the new license key you have selected.
3. Click **Activate** to activate the new license key; it is copied to the instance's *install-dir*/mgr directory as iris.key, overwriting the previous license key (if any). A confirmation dialog reminds you to restart the instance if required, and warns you if the new license enables fewer features than the current license.

Instances can be configured to request a license key from the license server by using the LicenseID property of Config.Startup. At instance startup, if there is no iris.key file present and the LicenseID has been defined, then the instance requests and activates the license key from the license server.

Note: The same LicenseID must be in the license key file, as well as defined on the instance that needs to download a license

In general there is no need to restart the instance, but there are constraints when upgrading a license key. Automatic activation of the new key does not occur if you change license types from Power Unit to any other type; this should be a rare event.

Another constraint is the amount of memory the license upgrade consumes from the shared memory heap (*gmheap*) space. If *gmheap* space is not available, the number of license table entries cannot be expanded. If insufficient *gmheap* space is available for a license upgrade, a message is written to the messages log. You can increase the size of the gmheap setting from the **Advanced Memory Settings** page (**System Administration** > **Configuration** > **Advanced Memory Settings**).

If the new license key consumes at least 1000 64 KB pages more *gmheap* space than the existing key, the InterSystems IRIS instance must be restarted to fully activate the new license key. This situation is rarely encountered, since each page represents at least 227 licenses.

14.6 Updating a License Key

To update a license key, replace the key file in the KeyDirectory and run **ReloadKeys^%SYS.LICENSE**. The License Monitor on each instance (**^LMFMON**) checks every 30 minutes to see if there is a different key for the configured LicenseID, and if so, tries to perform an upgrade.

Note: While most upgrades succeed on a live instance, some conditions could require an instance be restarted. The License Monitor logs an error in this case, and does not try to upgrade the key again until the next day (to avoid logging repeated errors). An instance restart loads the new key at startup.

14.7 License Troubleshooting

If only one user can log in after entering your license and restarting InterSystems IRIS, use the Management Portal to investigate. The **License Usage** page (**System Operation** > **License Usage**) shows how many processes are running when you select **Usage by Process**. You can also use the Portal to display license information from the **License Key** page (**System**

Administration > **Licensing** > **License Key**), as described in Activating a License Key. If the key is not valid, the **CustomerName** field contains an explanation.

You can also check the license error messages in the messages log and System Monitor log, which can be viewed in the Portal on the **Messages Log** page (**System Operation** > **System Logs** > **Messages Log**) and **System Monitor Log** page (**System Operation** > **System Logs** > **System Monitor Log**), respectively (see Monitoring Log Files). System Monitor writes license expiration warnings and alerts to these logs, while Health Monitor writes license acquisition alerts and warnings. When the license limit is exceeded, alerts are written to the messages log by the licensing module. In Application Monitor, you can configure license metric-based alerts to send email notifications or call notification methods. See Using System Monitor for more information.

When your license is nearing its expiration date, you will get warnings in the messages log. Your license is valid until the end of the day it expires on. For example, if your license has an expiration date of November 30, 2022, it will be valid until the end of November 30, 2022, but it will no longer work on December 1, 2022.

$System.License.Help displays a list of methods you can use to troubleshoot license problems:

ObjectScript

```
 Do $System.License.Help()
```

This document describes many of these methods.

14.7.1 Administrator Terminal Session

Several problem can prevent you from obtaining a Terminal session. This can happen when InterSystems IRIS fails to start properly and enters *single user mode*, or simply when there are no licenses available. In these cases you may need to create an administrator terminal session, which uses a special license to allow you to resolve the problem.

The command to open an administrator session differs for Windows and for UNIX®, Linux, and macOS.

14.7.1.1 Administrator Session on Windows

Using the command prompt, navigate to *install-dir*\bin. Then, execute the following command as an administrator:

```
irisdb -s<install-dir>\mgr -B
```

This runs the InterSystems IRIS executable from the InterSystems IRIS installation bin directory (*install-dir*\bin), indicates the pathname of *install-dir*\mgr (using the `-s` argument), and inhibits all logins except one emergency login (using the `-B` argument).

As an example, with an instance named *MyIRIS* located in the default directory, the command would look like:

```
c:\InterSystems\MyIRIS\bin>irisdb -sc:\InterSystems\MyIRIS\mgr -B
```

14.7.1.2 Administrator Session on UNIX®, Linux, and macOS

Using the command prompt, navigate to the *install-dir*/bin directory. Then, execute the following command:

```
iris terminal <instance-name> -B
```

As an example, with an instance named *MyIRIS* installed in the default directory, the command would look like:

```
User:/InterSystems/MyIRIS/bin$ iris terminal MyIRIS -B
```

14.8 Upgrading a License from the Operating System Command Line

The **%SYSTEM.License.Upgrade()** method activates a new license key that has been copied to the *installdir*\mgr directory. If all license units are consumed by users, preventing you from opening a Terminal window, you can run this method from the command line to activate a new license key with a greater capacity, as follows:

```
iris terminal <instancename> -U %SYS '##Class(%SYSTEM.License).Upgrade()'
```

For more information on the **iris** command, see Connecting to an InterSystems IRIS Instance.

14.9 Determining License Capacity and Usage

How does one know how many licenses have been used, and by whom? The %SYSTEM.License class provides an interface to the InterSystems IRIS license application programming interface (API) and presents a number of methods and related queries that you can use to query license capacity and current use.

You can run a number of licensing queries using the RunQuery method of the %Library.%ResultSet class. For example:

ObjectScript

```
do ##class(%ResultSet).RunQuery("%SYSTEM.License","Summary")
```

You can view the output of these queries from the **License Usage** page of the Management Portal (**System Operation** > **License Usage**), as detailed in the following table:

Link on License Usage Page	License Query
Summary	**Summary()** — returns license usage summary, as displayed by **$System.License.ShowSummary**.
Usage by Process	**ProcessList()** — returns license use by the operating system process identifier (PID), as displayed by **$System.License.DumpLocalPID**.
Usage by User	**UserList()** — returns license use by User ID.
Distributed License Usage	**AllKeyConnectionList()** — returns current distributed license usage sorted by users. (This is disabled when no license server is connected.)

You can also use the following class methods from the %SYSTEM.License class to display information, or dump the license database to a file:

$System.License.CKEY displays the key. This subroutine is called by the **^CKEY** program which is retained for compatibility:

ObjectScript

```
Do $System.License.CKEY()
```

$System.License.ShowCounts summarizes license use tracked in shared memory on the local system:

ObjectScript

```
Do $System.License.ShowCounts()
```

$System.License.ShowServer displays the active license server address and port:

ObjectScript

```
Do $System.License.ShowServer()
```

If you have developed REST based applications, your licenses will be consumed with use. To prevent this from happening, configure the number of Web Gateway connections that can be made. From the Management Portal in the Web Gateway Management section:

1. Navigate to **Server Access**.
2. Select **State-less Parameters**.
3. Set the **Maximum** to a number 2 or 3 less than the license to allow for server-side logins.

Note: Depending on the server side needs of the application you will need to adjust this.

By doing this when all the available connections are busy, new requests will queue up rather than being rejected. You will not see a rejection due to license counts being exceeded. As volume grows, the response time for the client slows down. That would be the indication that you need to buy more licenses.

The next sections describe several other methods that show license information:

- Methods to Show Local License Information
- Methods to Show License Server Information

14.10 Methods to Show Local License Information

The subroutines listed below dump the contents of license tables contained locally in instance shared memory. In general, they identify the client:

$System.License.DumpLocalAll dumps all local license table entries to the all.dmp file in the current directory:

ObjectScript

```
Do $System.License.DumpLocalAll()
```

An example of the contents of the all.dmp file:

```
License Capacity = 5, Current use = 2, Units Remaining = 3

   0)  User ID = 127.0.0.1, Connections = 2, CSP Count = 0, Time active = 90
   1)  User ID = 192.9.202.81, Connections = 1, CSP Count = 0, Time active = 49
   2)  free
   3)  free
   4)  free
```

$System.License.DumpLocalInUse dumps all local license table entries in use to the inuse.dmp file in the current directory:

ObjectScript

```
Do $System.License.DumpLocalInUse()
```

An example of the contents of the inuse.dmp file:

```
License Capacity = 5, Current use = 2, Units Remaining = 3
```

$System.License.DumpLocalPID dumps local license table use by process ID to the piduse.dmp file in the current directory:

ObjectScript

```
 Do $System.License.DumpLocalPID()
```

An example of the contents of the piduse.dmp file:

```
PID     Process LID     Type    Con     MaxCon  CSPCon  LU      Active  Grace

592     System                  0       0       0       0       0       0
2816    System                  0       0       0       0       0       0
688     System                  0       0       0       0       0       0
```

14.11 Methods to Show License Server Information

The following subroutines dump the contents of license tables maintained by the license server. The output files are in the indicated directory on the host where the active license server is running.

$System.License.ShowSummary displays a summary of license information at the license server. The `Distributed license use` section presents a collective view of license use for all InterSystems IRIS instances currently supported by the license server. The `Local license use` section presents a view of license use for the single InterSystems IRIS instance in which the program is run:

ObjectScript

```
 Do $System.License.ShowSummary()
```

$System.License.DumpServer dumps the license server database information relating to the server from which you run this routine to the file dumpserver.txt on the host running the license server:

ObjectScript

```
 Do $System.License.DumpServer()
```

$System.License.DumpServers dumps the license server database information for all known servers to the file dumpservers.txt on the host running the license server:

ObjectScript

```
 Do $System.License.DumpServers()
```

$System.License.DumpKey dumps the key used by this instance and instances that share it to the file dumpkey.txt on the host running the license server:

ObjectScript

```
 Do $System.License.DumpKey()
```

$System.License.DumpKeys dumps all keys, showing the instances and clients using them to the file dumpkeys.txton the host running the license server:

ObjectScript

```
Do $System.License.DumpKeys()
```

Note: Be aware that the information displayed by the local license methods is more up-to-date than the information shown by the license server methods; the license server is only updated periodically, while the local data is real time.

It is possible to exceed the license limit temporarily because login is controlled locally, but the license server enforces the limit. Each instance permits or denies logins based on its local license table which is maintained in instance shared memory. Each instance sends periodic updates to the license server describing changes to the local license tables. If the combined license use of all instances exceeds the limit, the license server sends a negative acknowledgment to update messages from each instance.

This negative acknowledgment causes each instance to refuse new logins because no additional license units are available. A login is considered new when the license user ID of the InterSystems IRIS process attempting to start does not match the license user ID of any current process. This state persists until the combined use by all instances falls below the authorized limit, at which point the license server begins sending positive acknowledgments in response to instance updates. The individual instances then allow new logins.

14.12 Identifying Users

The InterSystems IRIS licensing system attempts to identify distinct users and to allocate one license unit per user. A user is identified by a license user ID, which can be an IP address, a username, a web session ID, or some other identifier depending on how the user connects.

Multiple processes started by or for a single user share a license unit up to the maximum number of processes per user. If the number of processes exceeds this maximum, a transition occurs and InterSystems IRIS begins allocating one license unit per process for that user ID. The system assumes that if the number of processes associated with a user ID exceeds the maximum, multiple users are accessing InterSystems IRIS through an intermediary (for example, a firewall system), so additional license units are required. (Processes started by the **Job** command are counted under the user ID invoking the command.)

Even if the number of processes under the user ID drops back under the maximum, InterSystems IRIS continues to allocate one license unit per process for that user ID. Only when all connections by the user ID are closed and there are no more processes under the user ID does license allocation reset to one unit for that user ID.

Applications that identify their users by name eliminate problems associated with using a default user ID based on client IP address, web session ID, or other connection-derived user ID.

For example, when firewall or terminal server software is used, InterSystems IRIS cannot differentiate among connecting users, so it falls back on the maximum-connection transition rule. Using mixed connections from the same client also makes it impossible to count users appropriately using automatic ID creation.

When the username serves as the license identifier, these problems disappear. The importance of accurate user identification is expected to grow as organizations implement new access and audit requirements. Using the user identity to control license compliance is a natural corollary to this trend.

This section covers the following topics:

- License Logins
- Username Licensing
- License Login Special Cases

14.12.1 License Logins

There are two modes of license login: automatic and explicit. Automatic login is the default. The licensing system attempts to identify the IP address of the client and uses it as the license user ID. This works well when clients connect directly to the server using IP. It does not work well if a firewall intervenes between the client and the server; all clients appear to have the same IP address. When a terminal server is used with the telnet protocol, automatic login cannot differentiate among users because InterSystems IRIS sees a single IP address for all terminal server ports. Since all connections originate from the same address, all connections have the same user ID. If users connect through a firewall or use the telnet transport from terminal servers, use explicit logins.

When IP is not used as the network transport, the IP address is not available for use as a license user ID. In these cases, the licensing system uses a variety of other sources as the license user ID. Batch processes started by the **at** daemon on UNIX®/Linux systems pose another special case. Such processes do not share a license unit because they are not associated with a user. For these processes, the process ID is used as the license identifier.

When you select explicit login, InterSystems IRIS does not attempt automatic user ID detection. The application must explicitly call the **$System.License.Login(UserIdentifier)** method to supply the license user ID and acquire a license.

Enable explicit login by calling the **$System.License.DeferUserIdentification([0 or 1])** function. You can make this call from the `SYSTEM` entry point in the **^%ZSTART** routine at instance startup. If the argument value is `1`, license acquisition is deferred at login, so an explicit login can be performed. If the argument value is `0`, license acquisition is automatic at process startup.

When you defer login you must call the license login method immediately. A process that has not performed a license login pauses after its first 4000 ObjectScript commands, and then every 1000 ObjectScript commands after that.

Use an explicit login for any case that automatic login does not handle. It is important to remember that, even if automatic login is configured, it is always possible to call **$System.License.Login(UserIdentifier)** to use explicit user identification for licensing purposes.

14.12.2 Username Licensing

You can use the value of *$USERNAME* to identify users for licensing. This enables more accurate counting in situations where you cannot use only the IP address to reliably identify distinct users.

You modify how you specify the license user ID using the **$SYSTEM.License.UserNameLicensing()** method of the %SYSTEM.License class. By default, InterSystems IRIS uses the client IP address to identify a user to the license tracking subsystem. If you installed InterSystems IRIS with higher than Minimal initial security settings, each process has a user ID (*$USERNAME*). You can call the **$SYSTEM.License.UserNameLicensing()** system method to make the InterSystems IRIS license subsystem use *$USERNAME* as the license user identifier.

The **$SYSTEM.License.UserNameLicensing()** method modifies the system state. You can also call it from **SYSTEM^%ZSTART** to enable username licensing at instance startup. The method has the following functions:

- **$SYSTEM.License.UserNameLicensing(1)** — enables *$USERNAME* based licensing and returns the previous state.
- **$SYSTEM.License.UserNameLicensing(0)** — disables *$USERNAME* based licensing and returns the previous state.
- **$SYSTEM.License.UserNameLicensing()** — returns the current state. May return an error if called with an argument for license types that use special login rules.

For example, the following displays whether username licensing is currently enabled or disabled:

ObjectScript

```
 Write " Username Licensing",!
 Write " 1-enabled, 0-disabled",!
 Write $SYSTEM.License.UserNameLicensing(),!
```

The following example enables, then disables username licensing:

ObjectScript

```
Set RC=$SYSTEM.License.UserNameLicensing(1)
Write RC,!
Set RC=$SYSTEM.License.UserNameLicensing(0)
Write RC
```

See the $USERNAME special variable entry in the *ObjectScript Reference* for more information.

14.12.3 License Login Special Considerations

Bear in mind the following special considerations concerning license logins:

- CSP connections are a special case for logins. InterSystems strongly recommends that CSP applications use the %CSP.Session equivalent method, **%CSP.Session.Login**, to identify a user for licensing purposes. If they do not, the web session ID is used as the license user ID. Each session consumes a license unit, which in many cases is unsuitable. For example, a user can have several browser windows open concurrently. Alternatively, a user can connect via several pathways. In this case, you can use the %CSP.Session method, **%CSP.Session.Login(username, password)** to perform an explicit license login for the session.

  Note: When a CSP session ends (from a logout or timeout) and the user has visited only one page, CSP does not immediately release the license. Instead, CSP reserves the license for that user for a *grace period* of up to 10 minutes.

- Anonymous SOAP and REST requests (that is, requests that do not require Instance Authentications) consume a license unit for minimum of 10 seconds. However, any SOAP or REST request that identifies the user requires a license because it is considered a “user request.”
- InterSystems IRIS does not distinguish background processes and count them differently. If a user process starts another process, that child process counts as one more against the user’s overall maximum limit of processes.
- Each task created using the **New Task** page (**System Operation** > **Task Manager** > **New Task**) (see Using the Task Manager) consumes a license unit, with the license user ID based on the InterSystems IRIS username specified by the **Run task as this user** selector and the loopback IP address, **127.0.0.1**, which is converted to the host IP address. This ensures that tasks running as a given user on different hosts are counted together against the maximum limit of processes for that user discussed in Identifying Users.
- Processes started by the user startup routines (**^%ZSTART** and **^ZMIRROR**) are another special case. The process running the routine has no parent process. Therefore, a login is performed for the user ID, *User Startup*, before the routine is called. Processes started by the **Job** command from the routine have this user ID. If you prefer, you can call **$System.License.Login(UserId)** from the routine to change the user ID. This procedure means that the routine can start as many as one less than *maxconn* background processes and only consume one license. If, according to the license terms and conditions, these processes should have a separate license (for example if they drive a piece of laboratory equipment that requires a separate license), you are required to call **$System.License.Login(UserId)** to obtain a license for an appropriate user ID.

14.13 Application Licensing

Application licensing enables InterSystems application partners to take advantage of the InterSystems licensing capabilities for their own licensing purposes. InterSystems IRIS manages customer application licenses just as it does its own application licenses, maintaining usage counts and acquiring and returning user licenses as needed. Application licenses consumed by a process or a web session are automatically released along with the InterSystems IRIS license consumed by the process or session when a process exits, halts or is deleted from the process table, or when a web session times out or is deleted.

An application license is simply a file in standard .ini format, or a section of such a file, containing a section header identifying the application and some number of *keyword*=*value* pairs, unique within the license, representing the features licensed. Any correctly formatted application license can be loaded into InterSystems IRIS by an application at run time.

The application licensing API includes methods and queries that enable applications to consume and return licenses on behalf of a user and programs to obtain information about application and feature licensing, including the number of licenses in use and still available.

14.13.1 Loading an Application License

Any application license can be loaded and activated by an InterSystems IRIS instance at application run time using the **$SYSTEM.License.LoadAppLicenseFile** method, which is documented in the %SYSTEM.License class reference (see Application Licensing API). An application license loaded in this manner is not associated with the active InterSystems IRIS license, but is tracked independently by the InterSystems IRIS instance.

Each application license is contained in a section beginning with `[`*AppName*`]`; the application name (*AppName*) *cannot contain a period (.)*. The remainder of the license consists of a sequence of non-repeating *keyword*=*value* pairs representing the features licensed. See the **$SYSTEM.License.LoadAppLicenseFile** method documentation for more information about the required format.

In the following sample application license, the customer uses *keyword*=*value* pairs to limit the number of licensed users for several application features and enable the Extended Lab Reports feature for all users.

```
[BestLabApplication]
Lab Users=50
Lab Administrators=2
Lab Devices=5
Extended Lab Reports=Enabled
```

An application license is not protected from tampering by InterSystems IRIS, but it can be protected by custom application code. For example, a checksum can be embedded in the keyword section and validated by the application prior to activation.

14.13.2 Application Licensing API

The %SYSTEM.License class provides the following methods for managing application licenses:

LoadAppLicenseFile()

Loads an application license from a file.

GetAppLicense()

Returns the contents of a currently loaded application license as a string of *keyword*=*value* pairs.

GetAppLicenseMaxUsed()

Retrieves the maximum number of consumed license units associated with an application license keyword since the license was loaded

GetAppNames()

Returns the names of applications currently licensed by InterSystems, InterSystems application partners, or both.

IsAppLicensed()

Indicates whether an application or feature is licensed.

ReturnApplicationLicense()

Releases a license unit for an application license and feature, including the license resource associated with the license user ID of the current process, or the session license ID in the case of CSP.

SetConnectionLimit()

Sets the maximum number of connections permitted per user on this instance. This enables a system administrator to prevent a user from consuming all available license units

ApplicationServerLogin()

Indicates whether the number of instances using a currently loaded application license exceeds the limit specified in the key.

GetAppLicenseValue()

Retrieves the value associated with an application license feature keyword.

GetAppLicenseMinimum()

Retrieves the minimum number of free license units associated with an application license feature keyword, where the value associated with the keyword is numeric.

GetAppLicenseFree()

Retrieves the number of free license units associated with an application license feature keyword, where the value associated with the keyword is numeric.

GetProcessAppsTaken()

Retrieves a **$List** of application license types taken for the Process PID or "" for none

ApplicationUserList()

Returns license and application license use by user ID.

15

Managing InterSystems IRIS on Windows

Managing an InterSystems IRIS® data platform instance on the Microsoft Windows platform is straightforward. You can perform most tasks using the Management Portal and the InterSystems IRIS launcher (also known as the cube). You can also control an InterSystems IRIS instance from a command prompt.

This topic uses *install-dir* to refer to the InterSystems IRIS installation directory—you can find the default directory in the Default Installation Directory section of the *Installation Guide*.

CAUTION: *Do not* use Windows file compression on InterSystems IRIS IRIS.DAT database files. (Files are compressed by right-clicking a file or folder in Windows Explorer and selecting **Properties**, then **Advanced**, then **Compress contents to save disk space**; once compressed a folder name or filename is rendered in blue in Windows Explorer.) If you compress a IRIS.DAT file, the instance to which it belongs will fail to start, with misleading errors.

15.1 Managing Access to the InterSystems IRIS Instance

This section discusses the following topics related to managing access to InterSystems IRIS:

- The InterSystems Service
- Restricting Access to the Installation Tree
- Changing the InterSystems Service Account

15.1.1 The InterSystems Service

All InterSystems IRIS jobs and process run from the InterSystems service, *InterSystems IRIS Controller for <instance-name>*. The permissions the InterSystems service has are determined by its associated Windows user account. When this is the local SYSTEM account, InterSystems IRIS has broad access to all files and permissions on the Windows system. To maintain a more secure and restrictive environment, you should select a Windows account for the InterSystems service that only has the needed privileges and access. See Windows User Accounts for more information.

In Normal and Locked Down installations, InterSystems IRIS creates two local user groups that grant access to the instance. When you specify a Windows user account for the InterSystems Service other than the default local SYSTEM account, InterSystems IRIS adds that Windows user account to each group. These groups are:

- **IRISServices**, which grants the privileges to start, stop, and control the InterSystems IRIS instance.

- **IRIS_Instance**_*instancename*, which grants access to the installation tree—the directory in which InterSystems IRIS is installed and all its subdirectories.

Note: Regarding these groups and their privileges:

- When the **IRISServices** group is created, it is granted the **`Replace a process level token`** and **`Adjust memory quotas for a process`** privileges. Do not remove these privileges.
- The **IRISServices** and **IRIS_Instance**_*instancename* groups may not grant all the permissions that InterSystems IRIS requires to perform certain actions. To ensure InterSystems IRIS has the needed access to all instance, journal, and log files that are outside the installation tree, grant the **IRIS_Instance**_*instancename* group full access to these files and the directories containing them. You may also grant this group additional permissions if necessary.

Typically you select the Windows account for *InterSystems IRIS Controller for <instance-name>* during installation, as described in the Windows User Accounts. To change the service account after installation, see Changing the InterSystems Service Account below.

15.1.2 Restricting Access to the Installation Tree

By default, any authenticated Windows users can access the installation tree, which may be undesirable. To remove the Windows access control entry (ACE) for authenticated users:

1. On some Windows platforms, for example Windows 10, it may be necessary to first remove inherited permissions. To do so, use the following command

   ```
   icacls <install-dir> /inheritance:d
   ```

2. Then remove the ACE for authenticated users as follows:

   ```
   icacls <install-dir> /remove "NT AUTHORITY\Authenticated Users"
   ```

3. Finally, use the following command to confirm there are no references to “Authenticated Users”:

   ```
   icacls <install-dir>
   ```

   This command lists access permissions on the install directory.

Now only users that are administrators or in the **IRIS_Instance**_*instancename* group should have access to the installation tree.

Important: If you do not do this, any user who can log in to the host Windows system can easily modify files, change settings, or disable the InterSystems IRIS instance entirely.

In some cases, you may want to give another Windows account access to the installation tree, in addition to the account used by the *InterSystems IRIS Controller for <instance-name>* service. This could, for example, include accounts running automated tasks, or accounts that log in to the Windows server directly to access InterSystems IRIS (through a local Terminal session or by invoking a custom callin executable). You can give any such account the needed access by adding it to the IRIS_Instance_*instancename* group.

15.1.3 Changing the InterSystems Service Account

Enter the following in the command line to change the Windows user account used for *InterSystems IRIS Controller for <instance-name>*, the InterSystems service:

```
<install-dir>\bin\IRISinstall.exe setserviceusername <instance-name> <username> <password>
```

This command changes the Windows user account to the one you specify. It also adds the user to the **IRISServices** and **IRIS_Instance_***instancename* groups, creating these groups if necessary. After running this command and restarting an InterSystems IRIS instance, the instance runs under the newly specified Windows user account.

15.2 InterSystems IRIS launcher

The primary InterSystems IRIS interface on Microsoft Windows platforms is the InterSystems IRIS launcher. From the InterSystems IRIS launcher, you can start all of the InterSystems IRIS configuration and management tools. You can also invoke each Launcher command from a shortcut or command line.

Correspondingly, you can initiate many of the InterSystems IRIS tools from the Windows program menu by pointing to **IRIS** folder and then to the **Start InterSystems IRIS** for the appropriate InterSystems IRIS instance name.

When you start InterSystems IRIS on a Windows-based system, the InterSystems IRIS launcher IR appears in the system tray of the taskbar.

When you click the InterSystems IRIS launcher, a menu appears with commands to use the ObjectScript utilities and programming environments.

The following table describes the commands available from the InterSystems IRIS launcher menu.

InterSystems IRIS launcher Command	Description
Getting Started	Displays links to tutorials, release notes, documentation, and other related information.
Start InterSystems IRIS	Starts the default instance specified in the square brackets after the menu item, for example **[ii2081]**. If the InterSystems IRIS server is already started, this option appears dimmed—it is unavailable. **Note:** For information about how to prevent an instance from starting automatically, see Memory and Startup Settings.
Stop InterSystems	Shuts down or restarts the local InterSystems IRIS instance. If the InterSystems IRIS server is stopped, this option appears dimmed—it is unavailable.
Studio	Creates, edits, deletes, and compiles InterSystems IRIS class definitions and routines. See Using Studio for more information.
Terminal	Invokes the command line interpreter for InterSystems IRIS. See Using the ObjectScript Shell for more information.
Management Portal	Performs common system management tasks. Creates databases and namespaces, and adjusts all InterSystems IRIS configuration settings. Displays classes, globals, and routines, and functions for managing each. Displays tables and views, perform queries and SQL management functions. See Using the Management Portal.
Documentation	Displays InterSystems IRIS online documentation.

InterSystems IRIS launcher Command	Description
Preferred Server [*server name*]	Shows a list of remote servers and maintains server connections by using the **Add/Edit** command on the submenu. The preferred server appears in brackets and has a check mark next to it in the server list. See Define a Remote Server Connection
About	Displays InterSystems IRIS version and build information.
Exit	Removes the **InterSystems IRIS launcher** icon from the system tray; this does not stop InterSystems IRIS. The Launcher reappears when the instance is rebooted.

15.3 Starting InterSystems IRIS

To start InterSystems IRIS, run the startup procedure at the system level. This procedure runs using either the default configuration file or a configuration file you specify.

Note: If you have any trouble starting InterSystems IRIS, view the messages.log file as described in Monitoring Log Files.

To start InterSystems IRIS on the Windows platform, select **Start InterSystems IRIS** from the InterSystems IRIS launcher. This starts the InterSystems IRIS instance using the specified configuration file. When InterSystems IRIS is not running, the **InterSystems IRIS launcher** icon appears dimmed.

If the InterSystems IRIS launcher is not in the system tray, from the Windows program menu select the **IRIS** folder and **Start InterSystems IRIS** for that instance. To return the Launcher to the system tray, go to the *install-dir*/bin directory and double-click the iristray.exe file.

Alternatively, you can enter these commands from the *install-dir*/bin directory in the **Open** box of the **Run** command on the **Start** menu. For example, to start the instance named MyIris from the MyIris\bin directory, enter the following command:

```
c:\MyIris\bin\iris start iris
```

These methods of starting InterSystems IRIS call the **iris start** command. See Controlling InterSystems IRIS from the Command Prompt for more options and information on the **iris** command.

15.4 Stopping InterSystems IRIS

Normally you leave your InterSystems IRIS system running. However, if your operating system requires a restart, stop InterSystems IRIS before you shut down your system. The maintenance tasks, such as backups and database repair utilities, do not require you to stop InterSystems IRIS.

From the InterSystems IRIS launcher menu click **Stop InterSystems** to shut down or restart the local InterSystems IRIS instance. By default, this option shuts down (or restarts) InterSystems IRIS immediately, using the default shutdown routine. However, it also provides options for setting a timer for a delayed shutdown, for running a user-defined shutdown routine, for broadcasting a warning message to users on the server, and for shutting down without failing over. You can run this same process from the Windows program menu. Select the **IRIS** folder and **Stop InterSystems IRIS** for that instance name. You cannot cancel a shutdown once the countdown reaches 0 and the shutdown procedures have started.

Important: InterSystems recommends that you run **Stop InterSystems IRIS** to shut down InterSystems IRIS to ensure that it closes properly.

These methods of stopping InterSystems IRIS call the **iris stop** command. See Controlling InterSystems IRIS Instances for more options and information on the **iris** command.

Remove Stop InterSystems Command

To prevent unintentional execution of the **Stop InterSystems** command, you can remove the command from the InterSystems IRIS launcher by deleting the irisstop.exe file from the *install-dir*/bin directory of the corresponding InterSystems IRIS instance. You can also remove the **Stop InterSystems** shortcut from the appropriate InterSystems IRIS instance from the **Start** menu. Point to **Programs** and the InterSystems IRIS instance name, then right-click **Stop InterSystems** and click **Delete**.

15.5 Help Information from the Command Prompt

You can control an InterSystems IRIS instance from the Windows command prompt by running the iris.exe program in the *install-dir*\bin directory. For information about the **iris** command, see Controlling InterSystems IRIS Instances.

To display the most current help information for this command, invoke **`iris help`** from the *install-dir*\bin directory. For example:

```
C:\MyIris\bin>iris help
```

The **`iris help`** command will display the most current help information in the terminal. You can also save help information to a file in the *intsall-dir*\bin directory by adding `>outputfilename` to the **`iris help`** command. For example:

```
C:\MyIris\bin>iris help>helpinformation
```

15.6 Connecting to InterSystems IRIS on the Command Line

You can log in to an InterSystems IRIS instance on the command line using the **iris terminal** command. See Connecting to an InterSystems IRIS Instance for more information on **iris terminal**.

16

Managing InterSystems IRIS on UNIX®, Linux, and macOS

This page describes specific administrative procedures on UNIX®, Linux, and macOS.

16.1 UNIX® Users, Groups and Permissions

Every InterSystems IRIS® data platform installation on a UNIX® platform has the following users and groups:

- *Root* — InterSystems IRIS must be installed by `root`, and some processing by InterSystems IRIS system daemons runs as `root`.
- *Owner of instance* — This user owns most installation files and has full control of the instance. If you install with Minimal initial security settings, `root` is the default owner; otherwise, you are prompted for the owner during installation.
- *Effective user for InterSystems IRIS superserver and its jobs* — All InterSystems IRIS processes spawned by the superserver to serve incoming requests run as this user; in addition jobs hosted by job server processes, task manager jobs, and user-defined startup routines (for example, ^%ZSTART) also run as this user. By default, this user is `irisusr`, but you can change the user during a custom installation.
- *Effective group for InterSystems IRIS processes* — All InterSystems IRIS processes automatically run as this group, which allows normal users, while inside InterSystems IRIS, to access InterSystems IRIS database and journal files to which they may not otherwise have been granted access; file permissions on these and other InterSystems IRIS files are set to allow this group to have appropriate access. On a secure system, only the *Effective user for InterSystems IRIS superserver and its jobs* should be a member of this group. By default, this group contains `irisusr`, but you can change the group during a custom installation.
- *Group allowed to start and stop instance* — This group, `root`, and the *Owner of instance* can start and stop InterSystems IRIS.

All journals and journal directories must have the group ownership set to the *Effective group for InterSystems IRIS processes* group and grant full permissions to that group (`rw` for journals, `rwx` for journal directories). The user who owns the journal and journal directories may vary depending on how they were created.

Journals and journal directories created within InterSystems IRIS are created with the appropriate permissions. However, if you move, copy or create journal directories or journals externally (via scripts or administrator action), you must ensure that the proper permissions are maintained. Failure to set the permissions properly may lead to unexpected and serious errors.

The following example assumes the *Effective group for InterSystems IRIS processes* is `irisusr` and the *Owner of instance* is `irisowner`, although the files may have different user ownership depending on the context in which they were created. For example:

```
journal directory    irisowner    irisusr    drwxrwxr-x
20170801.001         irisowner    irisusr    -rw-rw----
```

Note: These settings are maintained, in part, as the set of permissions on the executables within the *install-dir*/bin directory of the InterSystems IRIS installation. Relevant properties include: ownership, group, mode, set-uid, and set-gid bits. It is important that you do not modify these permissions when performing administrative tasks at the operating-system level.

16.1.1 Database and Database Directory Permissions

All databases and database directories must have the group ownership set to the *Effective group for InterSystems IRIS processes* group and grant full permissions to that group (`rw` for databases, `rwx` for database directories). The user who owns the databases and database directories may vary depending on how they were created.

Databases and database directories created within InterSystems IRIS are created with the appropriate permissions. However, if you move, copy or create database directories or databases externally (via scripts or administrator action), you must ensure that the proper permissions are maintained. Failure to set the permissions properly may lead to unexpected and serious errors.

The following example assumes the *Effective group for InterSystems IRIS processes* is `irisusr` and the *Owner of instance* is `irisowner`, although the files may have different user ownership depending on the context in which they were created:

```
dataset directory    irisowner    irisusr    drwxrwxr-x
IRIS.DAT            irisowner    irisusr    -rw-rw----
```

16.2 Startup on UNIX®

The InterSystems IRIS instance uses the following resources to control starting, stopping, and creating new processes:

- The iris.ids file in the *install-dir*\mgr directory.
- Shared memory.

16.2.1 Daemon Resource Locks

InterSystems IRIS uses advisory file locking to prevent multiple startups of the same instance on different machines. With advisory file locking, a single lock file (in this case, the file ilock in the *install-dir*/mgr directory) may be used to exclusively lock multiple resources. The Control Process, Write daemon, and Journal daemon each lock a separate section of the lock file. If this section of the ilock file is already locked, startup terminates. The locks held by the different daemons are called *Daemon Resource Locks.*

A file lock is held by a process until the process terminates. Thus if any lock is held, it indicates that some daemon process on some node is running. It does not indicate, however, whether or not the instance is healthy and running normally.

16.2.2 iris.ids File

The iris.ids file contains the name of the node where InterSystems IRIS was started. The existence of the iris.ids file acts as a flag to ObjectScript utilities and customer-written scripts, indicating whether or not the instance is up and running. This file is often ignored during startup. However, if an error occurs when iris.ids is being read, it will prevent InterSystems IRIS from starting up. In previous versions of InterSystems IRIS, the shared memory identifiers were also stored in the iris.ids file, but this is no longer the case.

16.2.3 Startup Sequence

To best understand the startup sequence, imagine that the instance can be run from two (2) different nodes (machines), node A and node B. The iris.ids file is visible to both nodes, as are the Daemon Resource Locks (for shared files). The shared memory itself, however, is visible only on the node on which it was created (that is, the node where you started InterSystems IRIS).

16.2.3.1 Step 1. Check the Status of the Instance

The startup routine runs **irisdb –cV** to find out the status of the instance. It first attempts to attach to shared memory for the instance:

- If there is no shared memory for the instance, a test is made for Daemon Resource Locks:
  - If no Daemon Resource Locks are held, the instance is reported down.
  - If Daemon Resource Locks are held, the instance is reported to be running on the node specified in the iris.ids file. If the iris.ids file does not exist, no information is available on where the daemons are running.

    Action: The user must run **iris stop** or **iris force** to halt the running instance on the appropriate node. This stops the daemons and deletes the iris.ids file.
- If the attach succeeded, the system is assumed to be up and running. This status is reported to the user. Startup halts.
- If an error displays indicating that startup cannot be completed because shared memory is still attached, wait a few minutes for the memory to be released. If the error persists, there may be an error related to the iris.shid file, which tracks the shared memory ID of InterSystems IRIS on all UNIX® platforms except macOS. Contact the InterSystems Worldwide Response Center (WRC) for support correcting the file.

16.2.3.2 Step 2. Start InterSystems IRIS

The InterSystems IRIS startup process is run. Checks are repeated to ensure that another startup is not competing for the startup resources:

- If Daemon Resource Locks are held, indicating one or more daemons are running on some node for this instance, InterSystems IRIS reports this and exits with an error. Startup halts.

  The node on which the daemons are running is unknown if the iris.ids file does not exist.

  Action: The user must assume that another startup has occurred on some node. To determine on which node the instance has started, examine the iris.ids file.

InterSystems IRIS continues startup.

16.3 Managing InterSystems IRIS

From the shell, a user with any user ID in the `sysmgr` group can run **iris** (see Controlling InterSystems IRIS Instances) which invokes InterSystems IRIS executables and scripts in the *install-dir*/bin directory. The following sections describe how to perform these management tasks on an InterSystems IRIS instance:

- Starting InterSystems IRIS

- Running InterSystems IRIS
- Stopping InterSystems IRIS

Important: The owner of the installation has full privileges to start and stop the instance, to perform system administration, and to run diagnostic programs for that instance.

Only the user ID that is the owner of the instance can and should run all diagnostic activities. This ensures that any files or resources created are owned by the owner of the instance and not root (which may make it impossible to access these resources by a nonroot user). For this reason, it is inadvisable for root to in any way administer an instance not owned by root (including starting and stopping the instance). A user running as root should only administer instances owned by root.

16.3.1 Starting InterSystems IRIS

To start InterSystems IRIS, run the startup procedure at the system level. This procedure activates either a default configuration file or a configuration file you specify.

Note: If you have any trouble starting InterSystems IRIS, view the messages.log file as described in Monitoring Log Files.

If you are not on the console machine, run Telnet and connect to the target machine where InterSystems IRIS is installed. Before you can start InterSystems IRIS on UNIX®, one of the following must be true:

- You are the superuser.
- You have signed on as the root user. (It is acceptable to **su** (superuser) to root while logged in from another account.)
- Your UNIX® group ID matches the group named during the InterSystems IRIS installation as having privileges to stop and start the system.

  See Installing InterSystems IRIS on UNIX® and Linux for information on specifying such privileges during installation.

Start InterSystems IRIS using the **iris** command:

```
iris start <instname>
```

where *instname* is the name of the InterSystems IRIS instance you want to start. See Controlling InterSystems IRIS Instances for more options and information.

From the shell, a user with any user ID in the `sysmgr` group can run **iris start**. This command verifies that the instance is not currently running on the current or another node, creates shared memory and basic InterSystems IRIS daemons, including multiple auxiliary write daemons (AWDs), runs the startup (**^STU**) routine, which creates additional daemons (for example, ECP daemon), and then allows user logins.

16.3.2 Running InterSystems IRIS

From the shell, a user with any user ID and any group ID (`anyuser:anygroup` in this example), can run **iris terminal** (see Connecting to an InterSystems IRIS Instance) which executes irisuxsession in the *install-dir*/bin directory.

Running as `anyuser:irisusr`, InterSystems IRIS runs its standard startup logic, including Kerberos negotiation, to identify a **$USERNAME** and a set of login roles. In many cases, this **$USERNAME** value is associated with the actual user who invoked **iris terminal**. Thus, while any user may run InterSystems IRIS, the activities of that user once in InterSystems IRIS are defined and limited by the security roles assigned to that user.

CAUTION: Do *not* enter InterSystems IRIS by invoking its executable directly from the *install-dir*/bin directory.

The InterSystems IRIS executable is not itself a setgid-executable. It is the responsibility of the **iris terminal** wrapper to set the group properly on behalf of the user entering InterSystems IRIS.

In addition to **iris terminal**, you may run InterSystems IRIS using the irisdb executable in the /usr/bin directory. This command calls **iris terminal <default-instance>**, setting permissions and opening the Terminal for the default InterSystems IRIS instance. To create the irisdb executable, designate a default instance by running the **iris default <instname>** function.

Note: On macOS 10.11 and later, **iris** and **iris terminal** are located in /usr/local/bin, with links from /usr/bin.

16.3.3 Stopping InterSystems IRIS

Normally you leave your InterSystems IRIS system running. However, if your operating system requires a restart, you should stop InterSystems IRIS before you shut down your system. The InterSystems IRIS maintenance tasks, such as backups and database repair utilities, do not require you to stop InterSystems IRIS.

To stop InterSystems IRIS on UNIX®, the same requirements exist as for starting InterSystems IRIS. One of the following must be true:

- You are the superuser.
- You have signed on as the root user. (It is acceptable to **su** (superuser) to root while logged in from another account.)
- Your UNIX® group ID matches the group named during the InterSystems IRIS installation as having privileges to stop and start the system.

To stop InterSystems IRIS, from the command line:

1. Use the **iris stop** command:

   ```
   iris stop <instname>
   ```

   where *instname* is the name of the InterSystems IRIS instance you want to stop. (See Controlling InterSystems IRIS Instances for more **iris** options and information.)

   CAUTION: You can stop InterSystems IRIS with the **iris force** command, but you should do so with caution because it may result in a loss of data.

2. This procedure invokes the InterSystems IRIS **SHUTDOWN** utility, which displays a status report. Check for active processes in the report to determine if the next step is necessary.
3. Should it be necessary, broadcast a message to any users on the system:

   ```
   Do you want to broadcast a message to anyone? No=> Yes
   Send a message to other terminals. Message => Please sign off
   Terminal => /dev/tty/06
   Terminal =>
   Message =>
   ```

4. After sending one message you can send others, until you respond to the `Message` prompt by pressing **Enter**.
5. When the system asks if you would like to see another system status, enter **`Yes`** to see one, or press **Enter** if you do not want another report.
6. If you answer **`Yes`**, when the system status displays again, identify any active terminals.
7. Confirm that you want to halt by answering **`Yes`**. If you answer **`No`**, the shutdown procedure quits and InterSystems IRIS continues running.

Note: On UNIX® platforms, when an InterSystems IRIS instance is stopped, restarted, or forced down, the instance will wait for all processes to detach from shared memory for a maximum of 30 seconds. After 30 seconds, the instance will close. If there are still processes attached to the shared memory after the instance has closed, restarting the instance will fail.

17

Connecting to Remote Servers

This page describes how to use the InterSystems IRIS® data platform Launcher to access remote servers, as well as how to use the InterSystems IRIS Server Manager to define server connections.

The Launcher and the Server Manager are provided only on Windows.

17.1 Launcher Submenus

For quick access to remote InterSystems IRIS servers, the InterSystems IRIS launcher provides the following options that do not require having OS-level permissions on the machine hosting the remote instance:

- The **Remote System Access** submenu, which provides access to the Terminal and the Management Portal for any instance defined in the InterSystems IRIS Server Manager.

  This submenu also provides access to Studio, the documentation, and the class reference, not specifically discussed here.

  Remote System Access also provides an option you can use to connect via Telnet.

- The **Preferred Server** submenu, which lists the instances defined in the InterSystems IRIS Server Manager and which provides options to let you manage that list.

To use the utilities on the **Remote System Access** submenu for a remote instance:

1. Define a remote server connection to add the server to the preferred server list.
2. Click the InterSystems IRIS Launcher and select **Remote System Access**.
3. Select a utility (for example, the Management Portal)
4. Select the server name.

When connecting to the Management Portal, the documentation, or the class reference, you can bookmark the URLs for future use, for quicker access.

17.2 Define a Remote Server Connection

To use the InterSystems IRIS Launcher utilities or other InterSystems IRIS applications on a remote server, the server must be on the connection list in the InterSystems IRIS Server Manager. This is a list of remote servers you have previously defined to which you can quickly connect. A remote server is defined with an IP address for a unique server and a TCP port, which is an instance of InterSystems IRIS on that server.

Important: The InterSystems IRIS superserver must be running on the remote machine and its port must be open on your firewall to use the InterSystems IRIS Launcher utilities on that system.

For security reasons, username and password are not stored with the remote connection information.

To define the remote server:

1. From the InterSystems IRIS Launcher menu, point to **Preferred Server**, and click **Add/Edit** to open the InterSystems IRIS Server Manager. (On Windows systems, you must have Administrator privileges to take this step.)
2. Click **Add** to open the **Add Connection** dialog box.
3. Fill in the fields as described in the following table and click **OK**. Each field is required unless otherwise indicated.

- **Server Name** — A descriptive phrase that identifies the server; it is what appears as a selection in the InterSystems IRIS Launcher.
- **IP Address** — The IP address, host name (if you have a DNS server) or the fully qualified domain name (FQDN) of the remote server. InterSystems IRIS accepts any legitimate name reference for the remote server.
- **Port** — The port number of the superserver. The default port number is `1972`.
- **Telnet Port** — The port number of the telnet connection. The default port number is `23`.
- **Web Server IP Address** — *Optional*. The IP address of the web server you wish to use to manage this InterSystems IRIS instance. Defaults to IP Address if not specified.

  If you plan to use a web server that exists on a separate machine from the InterSystems IRIS instance you are managing, enter the IP address of the web server machine in this field. Defaults to **IP Address** if not specified.
- **Web Server Port** — The port number of the web server. The default port number is `52773`.
- **CSP Server Instance** — *Optional*. The InterSystems IRIS instance name to which you want to connect if you are configuring one web server to connect to multiple InterSystems IRIS instances. See Specifying CSP Server Instance for more information.
- **HTTPS** — Whether or not links from the InterSystems IRIS Launcher use HTTPS.
- **Authentication Method** — Choose **Kerberos** or **Password**.

  See Kerberos Authentication for details on the use of these fields.

  See Instance Authentication for information about password-based authentication in InterSystems IRIS.
- **Connection Security Level** — For *Kerberos* [1] only. Choose either **Kerberos**, **Kerberos with Packet Integrity**, or **Kerberos with Encryption**.
- **Windows InterSystems IRIS Telnet Server** — For *Kerberos* only. Select this check box if you are defining a connection to a Windows server.
- **Service Principal Name** — For *Kerberos* only. This field is pre-filled with the recommended service principal name format "cache/<FQHN>" (*FQHN* is the fully qualified host name) for the IP address you enter.
- **Comment** — *Optional*. A description of the remote server.

17.3 Specifying CSP Server Instance

If you are configuring one web server to connect to multiple InterSystems IRIS instances, enter the InterSystems IRIS instance name to which you want to connect in the **CSP Server Instance** field.

For example, if you have an IIS web server installed on a Windows machine and you also install two instances of InterSystems IRIS, iris1 and iris2, by default you manage each instance with its own private Apache web server that is installed as part of InterSystems IRIS.

However, you can also manage both instances from the public IIS web server by changing the **Web Server Port** (`80` by default for IIS) and specifying iris1 and iris2 in the **CSP Server Instance** field when creating a server definition for each instance. This automatically creates virtual directories /iris1 and /iris2 on the public web server that point to the corresponding InterSystems IRIS instances.

When you enter a value in the **CSP Server Instance** field and select the Management Portal on the Launcher for this server, the URI is formed as follows:

```
http://<web srvr addr>:<web srvr port>/<csp srvr inst>/csp/sys/UtilHome.csp
```

This places the instance name before the `/csp/sys/UtilHome.csp` portion of the URI and generates the following URIs for the two instances in the example.

```
http://localhost:80/iris1/csp/sys/UtilHome.csp
http://localhost:80/iris2/csp/sys/UtilHome.csp
```

If you plan to use InterSystems IRIS development tools and CSP applications remotely or control multiple instances from one web server, perform the configuration steps described in the Advanced Web Server Configuration section.

17.4 Advanced Web Server Configuration

For some InterSystems IRIS features to function properly on a remote connection, you must configure some additional web server settings on the remote instance.

From the web server machine, point the Management Portal to the remote InterSystems IRIS server instance and configure the following settings from the **Startup Settings** page of the Portal (**System Administration** > **Configuration** > **Additional Settings** > **Startup**):

- **WebServerName** — The DNS name or IP address of the web server that is configured for use with InterSystems IRIS tools (equivalent to the **Web Server IP Address** setting).
- **WebServerPort** — The port number of the web server (equivalent to the **Web Server Port** setting).
- **WebServerURLPrefix** — The name of the InterSystems IRIS server instance (equivalent to the **CSP Server Instance** setting).

For more information on complex configurations CSP and remote web servers, see the sections for your platform in Web Gateway Configuration Guide.

17.5 Connecting via Telnet

You can also connect to a remote instance of InterSystems IRIS from a Telnet session:

1. Click the InterSystems IRIS Launcher and point to **Remote System Access**.
2. Click **InterSystems IRIS Telnet**, connect to the remote server, and log on to the InterSystems IRIS system with your username and password.

You can remotely log in to an InterSystems IRIS instance on any supported platform from a terminal running on a PC or from any workstation client capable of running Telnet. This client may have only utilities and not an InterSystems IRIS server instance. The version of InterSystems IRIS on the client machine in most cases, must be the same or a later version of the InterSystems IRIS system it manages.

Note: On the macOS platform you can also use **SSH** from a command prompt on Windows to connect to your macOS machine and then connect with the following command:

```
sudo /sbin/service telnet start
```

18

Manage InterSystems IRIS Instances: The iris Command

You can install and run multiple instances of InterSystems IRIS® data platform on a single host system. Each instance is a unique, independent InterSystems IRIS environment.

18.1 Options for Managing an Instance

There are many ways to connect to and manage an InterSystems IRIS instance, which may be one of several installed on a given system. Two of the most common methods are as follows:

- The Windows launcher

  Each InterSystems IRIS instance installed on a Windows system has its own launcher in the system tray, which among other options lets you:

  - Connect to the instance by opening the Management Portal, the Terminal, and the Studio developer client.
  - Start, stop and restart the instance.
  - Display the documentation.

  From the launcher, you can also manage multiple remote InterSystems IRIS instances, including but not limited to, running remote backups, editing configuration settings, and creating and compiling remote objects and routines. See Connecting to Remote Servers for more detailed information.

- The **iris** command

  Executing the **iris** command on the operating system command line gives you management access to an InterSystems IRIS instance, which among other options lets you:

  - Connect to the instance using the ObjectScript shell.
  - Start, stop, and restart the instance.
  - Display information about the instance, and about other instances installed on the system.

  The **iris** command is described in detail in Connecting to an InterSystems IRIS Instance and Controlling an InterSystems IRIS Instance.

To use the **iris** command on a remote server, use a Telnet or SSH client; to use it with a containerized instance, use it inside the container, or use the **docker exec** command to run it from outside the container.

18.2 Connecting to an Instance

This section describes how to connect to an instance and have access to it via a shell, specifically the ObjectScript shell. You can use this shell in any namespace of an InterSystems IRIS instance. The ObjectScript shell is often called the Terminal, although the Terminal is actually a Windows application that provides this shell. To open this shell for a running instance, use the command **iris terminal** *instname*, where *instname* is the name you gave the instance at installation. A containerized instance is typically named **`IRIS`**.

Note: See Using The Terminal (video) on the InterSystems online learning website.

Log in using one of the predefined user accounts, with the password you provided during installation, or an account you created. The prompt that displays indicates the login namespace, for example:

```
# iris terminal IRIS

Node: intersystems2588, Instance: IRIS27

Username: admin
Password: ********
USER>
```

To exit the Terminal and close the window, enter the command **halt**.

When using the **docker exec** command to open the Terminal for a containerized instance (as described in Interact Using the InterSystems Terminal), you are automatically logged in as **irisowner** and do not need to authenticate.

On a Windows system, you must execute the command from its location, the *install-dir*\bin directory of an InterSystems IRIS instance, or include the full path in the command, for example **c:\InterSystems\IRIS27\bin\iris terminal IRIS4**. You can execute the binary of a given instance to connect to that instance or another; the instance name is required either way.

18.3 Controlling an Instance

The **iris** command supports a number of functions beyond **terminal**, and is invoked in the format **iris** *function instname arguments*, where *instname* is the instance name that you chose during the installation and *arguments* depends on the function.

Important: The **iris help** command displays all the command functions and arguments; the file IRISHelp.html is in the *install-dir*\Help directory. Some functions of the **iris** command are not listed in this document, but are shown in the **help** display.

The **iris** command behaves differently depending on the platform, and is described in the following tables:

- The iris Command on Unix®, Linux, and macOS
- The iris Command on Windows

18.4 The iris Command on Unix®, Linux, and macOS

Note: The **iris** command often displays error information in a message box. You can suppress this message box by adding **quietly** as the final argument to the **iris** command, which runs the command non-interactively with minimal dialog. This argument is also useful with other commands, such as when you want to shut the instance down without having to confirm the command.

iris all

Lists summary information for all installed instances, one instance per line, as described below.

Note: If you need complete information, such as for parsing or reporting purposes, use **iris list**.

iris allw

Lists the same information for each instance as **iris all**, without wrapping long field values. Lines longer than 80 characters may result.

iris console *instname* [*arguments*]

Opens the ObjectScript shell command window.

Arguments: same arguments as **iris terminal**.

iris force *instname*

Forces down the instance.

iris help [*arguments*]

Displays the most recent information about the **iris** command.

Arguments: **start**, **stop**, **force** — Display function-specific help for the start, stop, and force functions.

iris list [*arguments*]

Displays information about the installed InterSystems IRIS instances, as described below.

Arguments: *instname* — Optionally specify an InterSystems IRIS instance name to display only information about that instance. For example, `iris list MyIRIS` displays only information about the MyIRIS instance.

iris mdx *instname*

Provides direct terminal access to the MDX shell by running ##class(%DeepSee.Shell).%Go().

iris merge *instname* [*arguments*]

Applies a configuration merge file to the instance, updating its CPF (see Automating Configuration of InterSystems IRIS with Configuration Merge).

Arguments: [*merge-file*], [*target-CPF*] — You can optionally specify the location of the merge file to apply, the location of the target CPF (the active CPF for the instance), or both. For example, `iris merge MyIRIS /tmp/merge.cpf /net/home/MyIRIS` applies the merge file /tmp/merge.cpf to the instance named **`MyIRIS`**, the active CPF of which is in /net/home/MyIRIS. If the merge file and/or target CPF are not specified, environment variables are used if they exist; for more information, see How do I reconfigure an existing instance using configuration merge?

iris python *instname*

Provides direct terminal access to the Python shell by running ##class(%SYS.Python).Shell().

Note: For this command to work you must install Python as described in Using Embedded Python.

iris qall

Lists the same information for each instance as **iris all**, except that long lines are truncated to 78 characters plus a terminating tilde (~).

iris qlist [*arguments*]

Similar to **iris list**, but with additional information. The output for each instance (described below) is given on a single line, with fields separated by carets (^).

Arguments: *instname* — Optionally specify an InterSystems IRIS instance name to display only information about that instance. For example, `iris qlist MyIRIS` displays only information about the MyIRIS instance.

iris rename *instname newname*

Renames the instance.

iris restart *instname* [*arguments*]

Restarts the instance; equivalent of **iris stop** *instname* **restart**

Arguments: **nofailover** — Specify this optional argument to prevent triggering a mirror failover.

iris sql *instname*

Provides direct terminal access to the SQL shell by running ##class(%SQL.Shell).%Go()

iris start *instname* [*arguments*]

Starts the instance.

Note: You may be prompted to start in Emergency Mode; if so, see Emergency Access for more information.

Arguments:

- *full CPF path* — By default, InterSystems IRIS reads certain settings from the iris.cpf file located in the *<install-dir>*/mgr directory. You may provide the full path to a different .cpf file to use instead.
- **nostu** — Starts the specified instance without running **^STU**.

iris stat *instname*

Retrieves the same system statistics as the **irisstat** utility (see Monitoring InterSystems IRIS Using the irisstat Utility).

iris stop *instname* [*arguments*]

Shuts down the instance.

Arguments:

- **restart** — Starts the instance after shutting it down.
- **nofailover** — Specify this optional argument to prevent triggering a mirror failover.

- **quietly** — As the final argument, shuts down the instance without requiring confirmation. (Can also be used to run other *iris* commands noninteractively.)

iris stopnoshut *instname* [*arguments*]

Shuts down the named instance without running user shutdown routines, by running **INTNOSHUT^SHUTDOWN**.

Note: Only the instance owner and **`irisusr`** can run **INTNOSHUT^SHUTDOWN** without logging in to the Terminal.

Arguments: **nofailover** — Specify this optional argument to prevent triggering a mirror failover.

iris terminal *instname* [*arguments*]

Opens the ObjectScript shell for the instance.

Arguments:

- **-B** — Enables system administrator emergency login (see Administrator Terminal Session).
- **-b** *partition_size* — Specifies the maximum partition size (in KB) for the process.
- **"[***label***[+***offset***]]^***routine***"** — Specifies the name of an ObjectScript program to run in user mode. In addition to the specified formats, you can pass parameter lists consisting of string and/or numeric literals, as well as omitted (void) parameters, as follows:
  - `"routine[([parameter-list])]"`
  - `"[label]^routine[([parameter-list])]"`
  - `"##CLASS(package.class).method[([parameter-list])]"`

  where, for example, `parameter-list` is specified in the form `"string literal",,-+-000123.45600E+07`, and omitted parameters are passed to the target as `$Data(parameter)=0`.

  Note: Whitespace and shell meta characters must be quoted in an operating-system dependent form.

- **-U** *namespace* — Specifies the login namespace.

  Note: The **-U** argument has no effect if you are starting InterSystems IRIS with a user account whose **Startup Namespace** is specified (see User Account Properties).

18.5 The iris Command on Windows

On Windows, you must run the **iris** command from the *install-dir*\bin directory (or include the full path with the command).

iris all

Lists summary information for all installed instances, one instance per line, as described below.

Note: If you need complete information, such as for parsing or reporting purposes, use **iris list**.

iris allw

Lists the same information for each instance as **iris all**, without wrapping long field values. Lines longer than 80 characters may result.

iris console *instname* [*arguments*]

Opens the ObjectScript shell command window.

Arguments: Same arguments as **iris terminal**.

iris force *instname*

Forces down the instance.

iris help

Displays the most recent information about the **iris** command.

iris list [*arguments*]

Displays information about the installed InterSystems IRIS instances, as described below.

Arguments: *instname* — Optionally specify an InterSystems IRIS instance name to display only information about that instance. For example, `iris list MyIRIS` displays only information about the MyIRIS instance.

iris qlist [*arguments*]

Similar to **iris list**, but with additional information. The output for each instance (described below) is given on a single line, with fields separated by carets (`^`).

Arguments: *instname* — Optionally specify an InterSystems IRIS instance name to display only information about that instance. For example, `iris qlist MyIRIS` displays only information about the MyIRIS instance.

iris restart *instname* [*arguments*]

Starts the instance after shutting it down.

Arguments: **/nofailover** — Specify this optional argument to prevent triggering a mirror failover.

iris run *instname* [*arguments*]

Runs InterSystems IRIS in programmer mode with no input/output device for $Principal.

Arguments: Same arguments as **iris terminal**.

iris runw *instname routine* [*arguments*]

Runs the named InterSystems IRIS routine in application mode with no input/output device for $Principal. When run from a batch script, the command waits for the InterSystems IRIS process to terminate before returning the exit code from the process.

Arguments: *namespace* — Runs the routine in the specified namespace.

Note: The *namespace* argument has no effect if you are starting InterSystems IRIS with a user account whose **Startup Namespace** is specified (see User Account Properties).

iris start *instname* [*arguments*]

Starts the instance.

Note: You may be prompted to start in “Emergency Mode;” if so, see Handling Emergency Situations for more information.

Arguments: *full CPF path* — By default, InterSystems IRIS reads certain settings from the iris.cpf file located in the *<install-dir>*/mgr directory. You may provide the full path to a different .cpf file to use instead.

iris startnostu *instname*

Starts the specified instance without running **^STU**.

iris stop *instname* [*arguments*]

Shuts down the instance.

Arguments: **/nofailover** — Specify this optional argument to prevent triggering a mirror failover.

iris stopnoshut *instname* [*arguments*]

Shuts down the named instance without running user shutdown routines, by running **INTNOSHUT^SHUTDOWN**.

Note: Only the instance owner and `irisusr` can run **INTNOSHUT^SHUTDOWN** without logging in to the Terminal.

Arguments: **/nofailover** — Specify this optional argument to prevent triggering a mirror failover.

iris stopstart *instname* [*arguments*]

Starts the instance after shutting it down.

Arguments: **/nofailover** — Specify this optional argument to prevent triggering a mirror failover.

iris terminal *instname* [*arguments*]

Opens the InterSystems Terminal (formally the ObjectScript shell) for the instance.

Arguments:

- *routine* — Runs the named InterSystems IRIS routine in application mode in the Terminal for $Principal.
- **"[***label***[+***offset***]]^***routine***"** — Specifies the name of an ObjectScript program to run in user mode. In addition to the specified formats, you can pass parameter lists consisting of string and/or numeric literals, as well as omitted (void) parameters, as follows:
  - `"routine[([parameter-list])]"`
  - `"[label]^routine[([parameter-list])]"`
  - `"##CLASS(package.class).method[([parameter-list])]"`

  where, for example, `parameter-list` is specified in the form `"string literal",,-+-000123.45600E+07`, and omitted parameters are passed to the target as `$Data(parameter)=0`.

  Note: Whitespace and shell meta characters must be quoted in an operating-system dependent form.

- *namespace* — Used with *routine*, runs the routine in the indicated namespace.

  Note: The *namespace* has no effect if you are starting InterSystems IRIS with a user account whose **Startup Namespace** is specified (see User Account Properties).

18.6 iris list, qlist, and all

This topic describes contains additional details about some of the **iris** functions.

iris all

Lists the following information about one or more InterSystems IRIS instance:

- Instance status, as follows
  - <blank> (status unavailable, logins disabled)
  - dn (down or has crashed)
  - up (running)
  - st (starting or stopping)
- Instance name
- InterSystems IRIS version
- Superserver port number
- Installation directory

iris list

Lists the following information about one or more InterSystems IRIS instances:

- Instance name (and installation type)
- Installation directory
- InterSystems IRIS version
- Pathname of the current Configuration Parameter File
- Superserver and web server port numbers
- Instance status, as follows
  - running
  - down
  - starting or stopping
  - incomplete start or stop, logins disabled
- Instance’s system health state, if running (see System Monitor Health State)
- Product name
- Mirror member type and status (if a mirror member) (see **%SYSTEM.Mirror.GetMemberType()** and **%SYSTEM.Mirror.GetMemberStatus()**)

iris qlist

Outputs the following information on a single line, separated by carets (^), for one or more InterSystems IRIS instance:

1. Instance name (and installation type)

2. Installation directory
3. InterSystems IRIS version
4. Instance status
5. Pathname of the current Configuration Parameter File, relative to the installation directory. Windows systems instead show the full path.
6. Superserver port number
7. Web server port number
8. JDBC Gateway port number
9. Instance’s system health state, if running (always blank on Windows)
10. Product name
11. Mirror member type (if a mirror member)
12. Mirror status (if a mirror member)
13. Data directory (if applicable)

18.7 Configuring Multiple Instances

You can install and simultaneously run multiple instances of InterSystems IRIS on a single machine. Install InterSystems IRIS as for a single installation, giving each instance a unique name, a unique installation directory, and a unique port number for the superserver, web server, and Telnet.

The special considerations for multiple instances are:

- Installing multiple instances is limited by components where only one exists on a system. For example, typically there is only one web server on a system; and as such, the InterSystems IRIS installation configures the Web Gateway for the most recent installation. InterSystems IRIS client components stored in the registry encounter the same issue. InterSystems IRIS stores its ODBC driver in the registry using one name for each. Currently, the last installation updates these components to point to the last instance installed.

  InterSystems makes an effort to move common components to a common directory that can be shared across InterSystems IRIS instances.
- Multiple instances can share the same multiserver key, but if they do, they must use the same license server or set of license servers. Each system running an instance of InterSystems IRIS under the auspices of one or more license servers must have a local copy of the authorizing license key file installed in every instance.
- Multiple instances can be networked.
- Protection is included against simultaneous database use (that is, each instance must have its own databases and cannot access or modify another instance’s databases).
- Each instance must have unique port numbers. See Set Port Numbers.

18.7.1 Set Port Numbers

For a standard, single instance of InterSystems IRIS, the superserver port number is `1972` by default. For multiple instances of InterSystems IRIS on a single machine, each must have a unique port number. During installation, subsequent instances are assigned a port if you choose to set it automatically, or you can manually enter port numbers during the installation. A standard installation sets the following port numbers for your InterSystems IRIS instance as follows:

- Superserver port number — `1972` if available. Otherwise, the port is set to `51773` or the first available subsequent number.
- Web server port number — `52773` or the first available subsequent number.
- Telnet port number — `23`

You most likely do not need to change the superserver or web server port numbers because of the way the InterSystems IRIS installation assigns them. However, you do need to assign each instance a unique Telnet port number. You can change the superserver port value after installation from the **Memory and Startup** page (**System Administration** > **Configuration** > **System Configuration** > **Memory and Startup**) of the Management Portal, and you can change the web server and Telnet port values after installation from the **Startup Settings** page (**System Administration** > **Configuration** > **Additional Settings** > **Startup**) and the **Telnet Settings** page (**System Administration** > **Configuration** > **Device Settings** > **Telnet Settings**), respectively, of the Management Portal.

Note: If you frequently use and manage multiple instances of InterSystems IRIS on one machine, verify that each is assigned a unique port number. The assignment of unique port numbers avoids confusion when using the InterSystems IRIS launcher tools on multiple instances from one Windows client.

A

Configuring Third-Party Software to Work in Conjunction with InterSystems Products

InterSystems products often run in environments alongside other providers, where interactions between our products and such tools can have deleterious effects. InterSystems guidance about optimal, reliable configurations for deployment presume that our products can be deployed without interference from third-party tools. For instance, InterSystems has observed that software for security, system monitoring, or virus scanning may impact the installation, performance, and functionality of our products. This is particularly true for tools, such as virus scanners, that directly interact with files that are part of or are used by InterSystems products.

InterSystems understands that customers face business, compliance, and other requirements that impact decisions about what software runs in a given environment and how such software is configured.

Virus scanner considerations:

1. To deliver virus-checked software, InterSystems products are delivered out of a sanitized environment to our customers, and InterSystems provides a checksum for verification.
2. Scanners may report false positives for InterSystems products during installation or at runtime. InterSystems has no control over the findings of third-party security products like virus scanners. Please consult with the vendor of the product you are using for documentation details and guidance.
3. It is possible that scanners may detect false positives for virus patterns on InterSystems processes, and they may respond with actions such as file isolation or even deletion. These actions can have potentially catastrophic impact, especially if performed while the instance is running. Files that are modified rapidly, such as databases, journals, etc., have a higher probability of presenting such patterns.
4. Near constant or very high frequency scans on large files (such as databases) very likely will have a considerable impact on the overall performance of InterSystems products.
5. In accordance with your own risk assessments, you may wish to adjust scanning for the following files and directories:
   - The WIJ file and the directory containing the WIJ file; this directory varies based on configuration settings. See the "Write Image Journaling and Recovery" page.
   - All database files (.DAT) and directories containing database files; these vary based on configuration settings. See the "Configuring Databases" page.
     – Any directory in which journal files are stored or processed; this directory varies based on configuration settings. See the "Journaling" page.

- Any other file or directory that is actively required for InterSystems IRIS® data platform to function; again, these vary based on configuration settings. Examples include the alternate journal directory (as described on the “Journaling” page), or any directory being used by a business service or production.

Long-term or permanent exclusions of files from virus scanners might increase the risk of getting “infected.”

CAUTION: EXCLUDING ITEMS FROM MALWARE SCANS MAY INTRODUCE VULNERABILITIES INTO PROTECTED DEVICES AND APPLICATIONS. THE CLIENT ASSUMES ALL RESPONSIBILITY FOR CONFIGURING MALWARE PROTECTION.

B

Feature Tracker Collects Usage Statistics

As part of ongoing efforts by InterSystems to improve its products in line with customer needs, InterSystems includes a software utility in InterSystems IRIS® data platform called Feature Tracker, which gathers statistics on software module usage. This topic describes Feature Tracker.

B.1 Why InterSystems Gathers Statistics

The statistics gathered by Feature Tracker record whether or not software modules are present and used in a given InterSystems IRIS instance. Feature Tracker sends this information to InterSystems weekly via https. These statistics help InterSystems plan development and support.

The information gathered does not include any application data.

B.2 Where Logs Are

If InterSystems IRIS is successful in sending data, it updates the file FeatureTracker.log in the <*installdir*>\mgr directory. This file contains a JSON-format copy of the data that was sent. Each entry has the following form:

```
"FT.<feature>":<value>,
```

If InterSystems IRIS is unable to send the data, it tries several more times. If it is still unsuccessful, it stops trying and tries again at the next regularly schedule time.

InterSystems IRIS updates the messages log file (messages.log) with both successful and unsuccessful attempts. An entry showing a successful send has the following form:

```
mm/dd/yy-hh:mm:ss 0 %SYS.Task.FeatureTracker transferred data to ats.intersystems.com
```

An entry showing an unsuccessful attempt to send has the following form:

```
mm/dd/yy-hh:mm:ss 1 %SYS.Task.FeatureTracker failed to transfer data
```

Transmitted data includes an encoded license key and host name, as well as the instance GUID.

B.3 What Statistics are Gathered

Feature Tracker logs whether the following software features are enabled:

- BitTrakCare
- BI User (Runtime) and BI Development
- C-Type License
- ExtremeNoUserLimit
- Healthshare, Healthshare Foundation
- IPNeutral
- Web License

Feature Tracker also logs the following information:

- Database (DB) Encryption: Whether it is enabled, and number of mounted encrypted databases available at snapshot time
- Mirroring: Whether this instance is connected to a mirror, number of failover members, number and type of async members, and number of mirror sets each async member is a member of
- System: Authorization, InterSystems IRIS version, operating system, host name, instance name, instance ID, and order ID

B.4 How to Deactivate Feature Tracker

Feature Tracker is enabled by default. You can deactivate it to prevent it from sending data to InterSystems (though it will still collect the data). To deactivate Feature Tracker, use the Task Manager as follows:

1. In the Management Portal, select **System Operation** > **Task Manager** > **Task Schedule**.
2. Locate the Feature Tracker line and select **Feature Tracker**.
3. On the displayed task detail page, select **Suspend**.
4. Answer the question `Do you want to reschedule task when task is supposed to run?` as appropriate.
5. Select **Perform this action now** to suspend the task.

If you upgrade InterSystems IRIS, the upgrade preserves the state of Feature Tracker. If the task was scheduled before the upgrade, it remains scheduled, and if the task was suspended, it remains suspended.