



Caché System Administration Guide

Version 2017.2
2020-06-25

Caché System Administration Guide
Caché Version 2017.2 2020-06-25
Copyright © 2020 InterSystems Corporation
All rights reserved.

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

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

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

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

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

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

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

Table of Contents

About This Book	1
1 Using the Management Portal	3
1.1 Starting the Management Portal	3
1.1.1 Management Portal Login Page	4
1.1.2 Management Portal Web Application Structure	4
1.2 Management Portal Overview	5
1.2.1 Management Portal Title Bar	5
1.2.2 Management Portal Ribbon	6
1.2.3 Management Portal Menu Column	7
1.2.4 Management Portal Navigation Pane	7
1.2.5 Management Portal Message Pane	10
1.2.6 System Overview Information	10
1.3 Management Portal Functional Overview	11
1.3.1 System Operations	11
1.3.2 System Explorer	12
1.3.3 System Administration	12
2 Configuring Caché	13
2.1 Configuring Data	15
2.2 Configuring Namespaces	16
2.2.1 Create/Modify a Namespace	16
2.2.2 Create a Namespace on an Ensemble Instance	17
2.2.3 Rename a Namespace or Modify Default Mappings	17
2.2.4 Add Global, Routine, and Package Mapping to a Namespace	18
2.2.5 Delete a Namespace	22
2.3 Configuring Databases	23
2.3.1 Database Considerations	23
2.3.2 Local Databases	25
2.3.3 Remote Databases	30
2.4 Configuring System Information	32
2.4.1 Memory and Startup Settings	32
2.4.2 IPv6 Support	33
2.5 Configuring Task Manager Email Settings	34
2.6 Configuring National Language Support (NLS)	35
2.6.1 Using the NLS Pages of the Management Portal	35
2.6.2 Using the NLS Class Packages	37
2.7 Configuring Cluster Settings	38
3 Managing Caché	39
3.1 Maintaining Local Databases	39
3.1.1 Displaying Free Space Information	40
3.1.2 Compacting a Database	42
3.1.3 Truncating a Database	43
3.1.4 Compacting Globals in a Database	44
3.1.5 Defragmenting Globals in a Database	45
3.2 Controlling Caché Processes	46
3.2.1 Display Process Information	47
3.2.2 Display Process Details	48

3.2.3 Broadcast Messages to the Terminals	50
3.2.4 Manage Refresh Interval	51
3.3 Using the Task Manager	51
3.3.1 New Task	52
3.3.2 On-demand Task	55
3.3.3 Upcoming Tasks	55
3.3.4 Task Schedule	55
3.3.5 Task History	56
3.3.6 Import Tasks	56
3.4 Using the Background Tasks Page	56
4 Managing Caché Licensing	57
4.1 Configuring Caché Licensing	57
4.1.1 Configuring License Servers	58
4.2 Activating a License Key	59
4.2.1 License Troubleshooting	59
4.2.2 Upgrading a License from the Operating System Command Line	60
4.3 Determining License Capacity and Usage	60
4.3.1 Methods to Show Local License Information	61
4.3.2 Methods to Show License Server Information	62
4.4 Identifying Users	63
4.4.1 License Logins	64
4.4.2 Username Licensing	64
4.4.3 License Login Special Considerations	65
4.5 Application Licensing	65
4.5.1 Loading an Application License	66
4.5.2 Application Licensing API	66
5 Using Caché on Windows	69
5.1 Caché Launcher	69
5.2 Starting Caché	71
5.3 Stopping Caché	71
5.4 Removing the Launcher from the System Tray	72
5.4.1 Temporarily	72
5.4.2 Permanently	72
5.5 Controlling Caché from the Command Prompt	72
5.6 Connecting to Caché on the Command Line	72
6 Using Caché on UNIX®, Linux, and macOS	73
6.1 UNIX® Users, Groups and Permissions	73
6.1.1 Database and Database Directory Permissions	74
6.2 Startup on UNIX®	74
6.2.1 Daemon Resource Locks	74
6.2.2 Cache.ids File	74
6.2.3 Startup Sequence	75
6.3 Managing Caché	75
6.3.1 Starting Caché	76
6.3.2 Running Caché	76
6.3.3 Stopping Caché	77
7 Connecting to Remote Servers	79
7.1 Define a Remote Server Connection	79
7.1.1 Using Optional Server Fields	81

7.2 Advanced Web Server Configuration	81
8 Using Multiple Instances of Caché	83
8.1 Caché Instances	83
8.2 Connecting to a Caché Instance	84
8.3 Controlling Caché Instances	85
8.4 Configuring Multiple Caché Instances	89
8.4.1 Set Port Numbers	89
Appendix A: Configuring Third-Party Software to Work in Conjunction with InterSystems Products	91
Appendix B: Feature Tracker Collects Usage Statistics	93
B.1 Why We Gather Statistics	93
B.2 Where Logs Are	93
B.3 What Statistics are Gathered	94
B.4 How to Deactivate Feature Tracker	94

List of Figures

Figure 1–1: Management Portal Home Page	5
Figure 1–2: Search Tool	7
Figure 1–3: Columns view	8
Figure 1–4: List View	9
Figure 1–5: Icon View	10
Figure 2–1: Sample Namespace Mapping	19
Figure 5–1: Caché Launcher Menu	70

List of Tables

Table 1–1: System Operation Tasks	11
Table 1–2: Database Management Tasks	12
Table 1–3: System Administration Tasks	12
Table 2–1: System Configuration Tasks	13
Table 2–2: Connectivity Tasks	14
Table 2–3: Additional Tasks	14
Table 3–1: Local Databases List Information	39
Table 3–2: Local Databases Freespace Information	41
Table 3–3: Process Management Functions	47
Table 3–4: Process Column Information	47
Table 3–5: General Information	48
Table 3–6: Client Application Details	49
Table 3–7: Execution Details	49
Table 7–1: Add Connection Input Fields	80
Table 8–1: Control Commands	83
Table 8–2: Caché csession Command and Arguments	85
Table 8–3: Caché ccontrol Command and Functions	86

About This Book

This book is an introduction to accessing and running a Caché system. Managing Caché is easy; whether running on a single PC or on a system of thousands of users, most of the information you need to get the system running is contained in this document.

Caché is available on a number of different operating systems. Certain administrative procedures and tasks may differ according to platform. Where this is the case, this document describes these differences. On all supported operating system-web server combinations, Caché provides the browser-based application to manage and administer your Caché system. You can also use the portal from a remote web server.

On Microsoft Windows systems, Caché also provides a set of graphical user interface (GUI) system utilities (available from either the Caché Cube in the Windows system tray or from the **Caché** submenu of the Windows **Programs** menu). These graphical utilities are client/server applications that can talk to the local Caché system or a remote Caché server regardless of platform. In addition, Caché provides a set of analogous, text-based utilities that can be invoked from the Terminal or used to manage a remote system using Telnet.

This book introduces the following topics:

- [Using the Management Portal](#)
- [Configuring Caché](#)
- [Managing Caché](#)
- [Managing Caché Licensing](#)
- [Using Caché on Windows](#)
- [Using Caché on UNIX®, Linux, and macOS](#)
- [Connecting to Remote Servers](#)
- [Using Multiple Instances of Caché](#)
- [Configuring Third-Party Software](#)
- [Feature Tracker Collects Usage Statistics](#)

For detailed information, see the [Table of Contents](#).

For general information, see [Using InterSystems Documentation](#).

1

Using the Management Portal

Caché lets you perform system administration and management tasks via a web application, the InterSystems Management Portal. An advantage of this approach is that Caché does not have to be installed on the system you use to manage an installation. Remote management of systems over a network, subject to access control established for the site, is now much easier. Cross-release compatibility issues are minimized because both the data and its formatting information come directly from the system being managed.

This chapter describes the following topics:

- [Starting the Management Portal](#)
- [Management Portal Overview](#)
- [Management Portal Functional Overview](#)

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 Caché Cube menu.
- In your web browser go directly to the Management Portal URI, 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 installed by Caché. For example, the default location on the local server of a single installation of Caché is <http://localhost:57772/csp/sys/UtilHome.csp> (where 57772 is the default Caché web server port number).

For information about the Management Portal, see [Management Portal Overview](#) in this chapter.

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

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:

You are logged out.
Please click here to login.

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 Title Bar](#), 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.

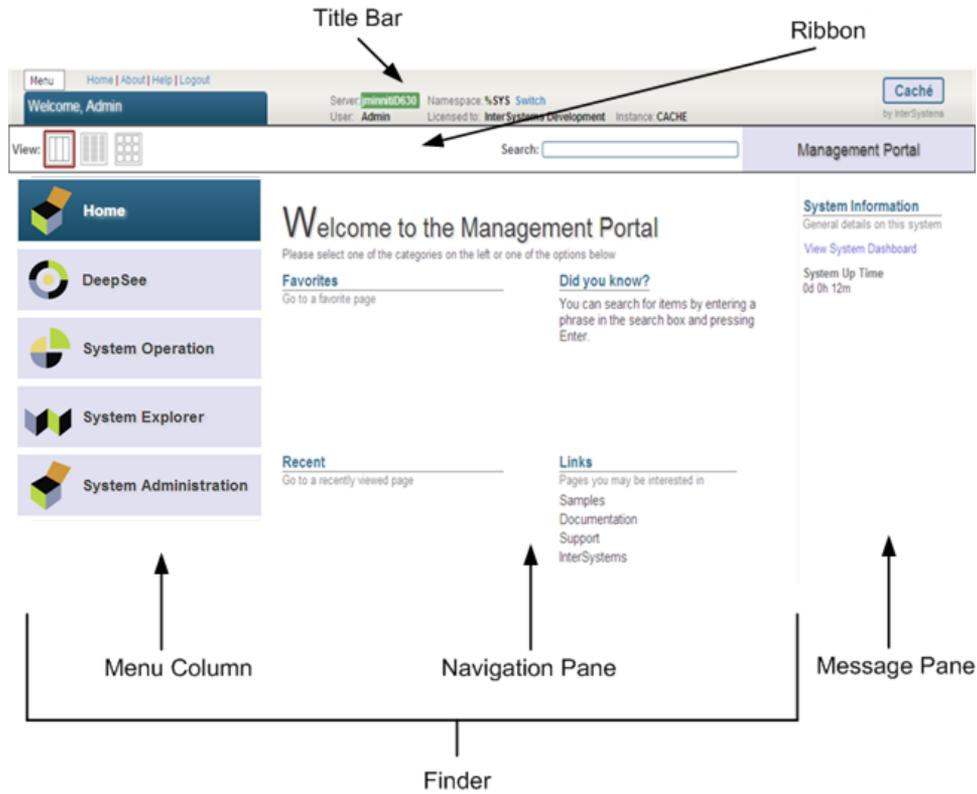
There are four types of user for the portal, which roughly correspond to predefined resources within Caché:

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 the various elements of Management Portal pages, as shown in the following figure.

Figure 1–1: Management Portal Home Page



The following sections describe each area of the Management Portal:

- [Management Portal Title Bar](#)
- [Management Portal Ribbon](#)
- [Management Portal Menu Column](#)
- [Management Portal Navigation Pane](#)
- [Management Portal Message Pane](#)

Note: Moving your cursor over a menu item displays a description of that item.

1.2.1 Management Portal Title Bar

The left side of the Management Portal title bar displays the following commands:

- **Menu** — Displays a list of common tasks based on the roles the user holds.
- **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.

- **Logout** — Logs you out and brings you to the **Login** page of the Management Portal.
- **Page name tab** — Displays the name of the current Management Portal page, for example **Local Databases > iKnow Configurations** or **Task Schedule**. When you edit an existing named element, the page name tab reads **Edit: *element_name***; for example, if you are editing the User database, the tab reads **Edit: User**. When you are creating an element, the tab reads **New *element***, for example **New Namespace**. When you have made changes to settings or element properties that have not yet been saved, an asterisk is appended to the tab's contents, for example **Journal Settings***. (You are always prompted for confirmation before navigating away from unsaved changes.)
- **Page Locator** — Displays all or most of the navigation path to the current page. Each page listed in the path is an active link, which you can use to return to a previously displayed submenu/list. For example, the location bar for the **Memory and Startup** page contains **System > Configuration > Memory and Startup**, which provides links to the **System Administration** menu option and the intervening **Configuration** menu option.

Note: The locator bar does not list every page in the path. You can always navigate to a page that is not included by clicking **Home** to return to the Management Portal home page and using the search tool, which is described in the [Management Portal Ribbon](#) section of this chapter.

The center part of the Management Portal title bar displays the following information:

- **Server** — The name of the server running Caché.
- **User** — The name of the user logged into the Management Portal.
- **Namespace** — The name of the namespace currently being used.

Note: To change the current namespace, click **Switch** and select the namespace in which you want to do your work.

- **Licensed to** — Customer name that appears in the license key information.
- **Instance** — The name of the Caché instance running on the server.

In addition, you can display a predefined **System Mode** label (for example, This is a Test System); for information, see [Memory and Startup Settings](#) in the “Configuring Caché” chapter of the *Caché System Administration Guide*.

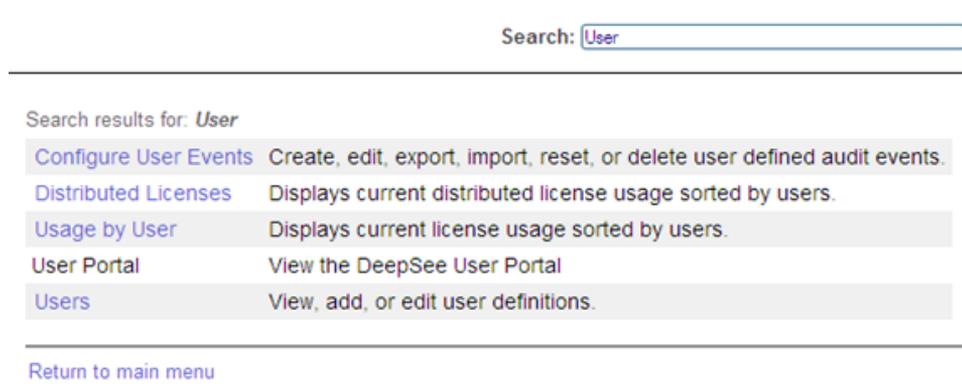
The right side of the Management Portal title bar displays the name of the product you are using.

1.2.2 Management Portal Ribbon

When you are on the home page, the Management Portal ribbon functions as follows:

- The left side of the ribbon displays buttons that let you specify how you want to display links in the Management Portal navigation pane (view modes); for more information, see the [Management Portal Navigation Pane](#) section of this chapter.
- The center part of the ribbon lets you [search](#) for a page in the Management Portal. When you specify a word that appears on one or more pages, 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/lists/icons.

Figure 1–2: Search Tool



- The right side of the ribbon reads **Management Portal** on a blue-gray background.

When you are on a page other than the home page, the Management Portal ribbon (if present) functions as follows:

- The left side of the ribbon includes buttons used to perform operations on the page, for example **Export > Import > Find > Replace > Delete > View > Classes**, and **Routines** on the **Globals** page. Some pages also include page refresh controls.
- The right side of the ribbon displays the page name on a background indicating the area of the portal to which it belongs—aquamarine for **System Operation**, green for **System Explorer**, or purple for **System Administration**.

1.2.3 Management Portal Menu Column

The Management Portal menu column links to the functional areas of Caché and InterSystems products (for example, DeepSee) that you have installed:

- **Home** — Displays the Management Portal home page.
- **DeepSee** — Invokes the InterSystems DeepSee web pages; see the “[Introduction to the DeepSee User Interfaces](#)” chapter of the *Getting Started with DeepSee* guide for more information.
- **System Operation** — System operation tasks; see [Management Portal Functions Overview](#) in this chapter for more information.
- **System Explorer** — Database management tasks; see [Management Portal Functions Overview](#) in this chapter for more information.
- **System Administration** — System management tasks; see [Management Portal Functions Overview](#) in this chapter for more information.
- **Ensemble** (Ensemble installation only) — Ensemble tasks; see the [Finding Information on Ensemble Menu Items](#) appendix of the *Managing Ensemble* guide for more information.

1.2.4 Management Portal Navigation Pane

The Management Portal navigation pane initially displays the following:

- Home page.
- View (Navigation modes) — Columns, lists, icons, and search tool that lets you navigate to a specified page.

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 the [Management Portal Ribbon](#) section in this chapter.

The home page contains the following sections:

- **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.
- **Did you know?** — Displays tips.
- **Recent** — Lists the most recent pages you have displayed since the last time Caché was started.
- **Links** — Links to pages you might want to visit.

Depending on the navigation mode you select in the Management Portal ribbon, the navigation path is displayed as follows:

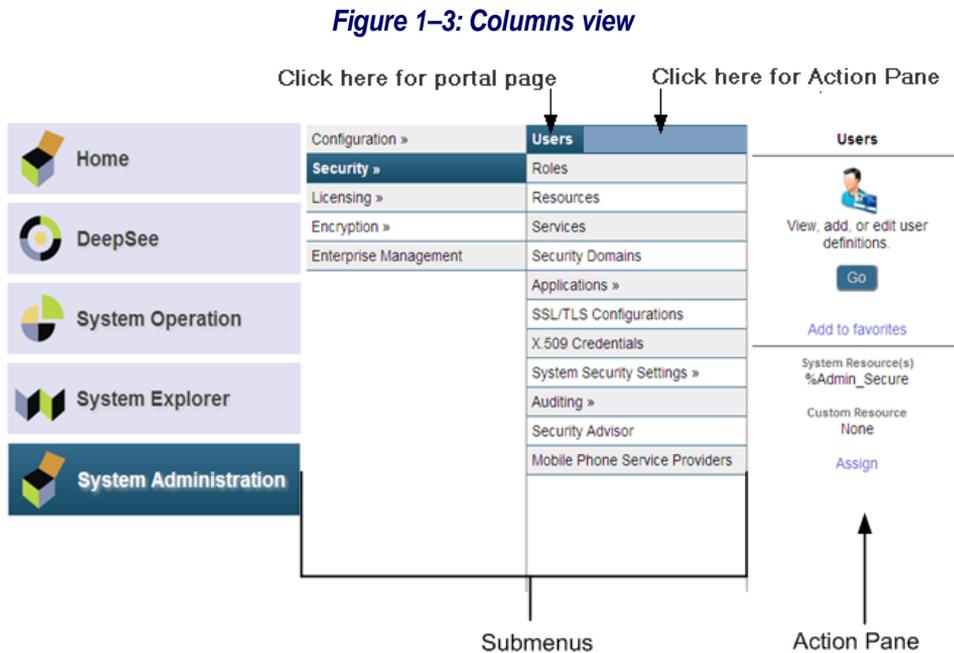
- **Columns view** — Displays page in columns; this is the default mode.

Note: This is the view you use to select favorites and assign custom resources; for information, see [Action Pane](#).

- **List view** — Displays page names in drop-down list.
- **Icon view** — Displays page names as labeled icons.

1.2.4.1 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.



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 Navigation Pane](#).
- View the system resource required to load the page and assign a custom resource, as described in [Using Custom Resources with the Management Portal](#) in the “Assets and Resources” chapter of the *Caché Security Administration Guide*.

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.

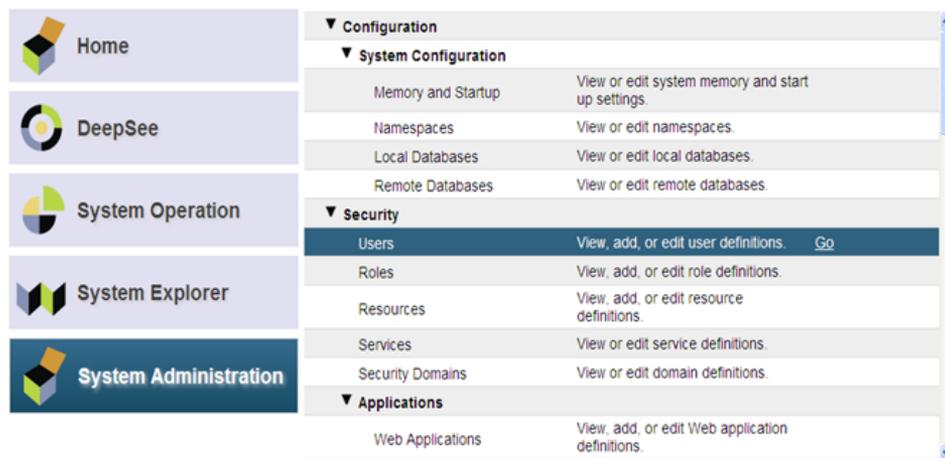
- Go to the destination page by clicking **Go**.

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

1.2.4.2 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**.

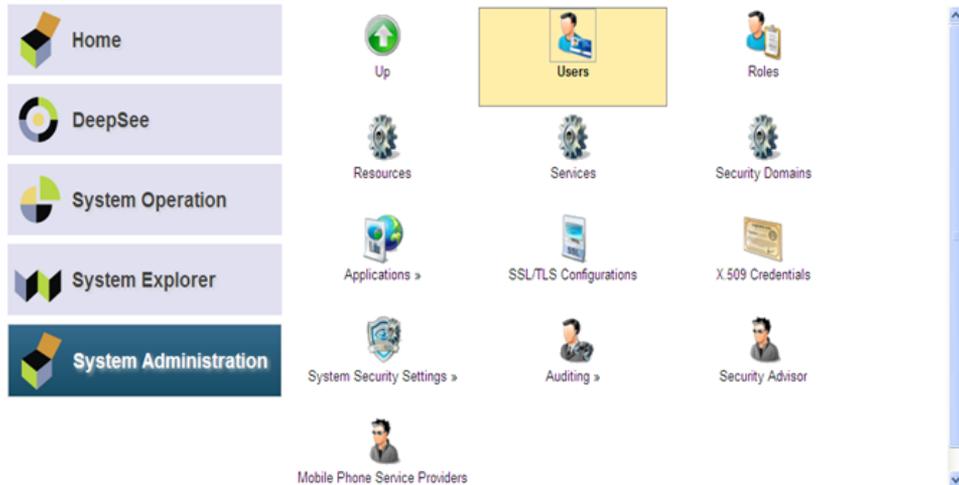
Figure 1–4: List View



1.2.4.3 Icon View

The “[icon view](#)” displays icons 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. To display the next level of icons or the destination page, double-click an icon.

Figure 1–5: Icon View



1.2.5 Management Portal Message Pane

The Management Portal message pane displays general system information, and provides a link to the system dashboard. For more information about the dashboard, see the “[Monitoring Caché Using the Management Portal](#)” chapter of the *Caché Monitoring Guide*.

If the instance is a mirror member, the mirror it belongs to, its member type, and its status are also displayed in the message pane, along with a link to the Mirror Monitor (see [Monitoring Mirrors](#) in the “Mirroring” chapter of the *Caché High Availability Guide*).

1.2.6 System Overview Information

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

- **Version** — Specific build information for this instance of Caché 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 Caché server is running.
- **Web Server Port** — Port number on which the private Caché Web server is running.
- **License Server Address/Port** — IP address of the Caché 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.
- **Time System Started** — Date and time this instance of Caché 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](#) section in the “Customizing the Caché System” chapter of *Caché Specialized System Tools and Utilities* and the [Using the NLS Pages of the Management Portal](#) section in the “Configuring Caché” chapter of this guide.

- **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 Management Portal Functional Overview

The Management Portal is divided into the following functional areas for configuring and managing Caché:

- [System Operations](#) — System operation tasks.
- [System Explorer](#) — Database management tasks.
- [System Administration](#) — System administration tasks.

The details of performing each task is contained in the appropriate Caché documentation. This document provides links to the appropriate section for each topic.

1.3.1 System Operations

The system operator tasks are divided into the categories displayed in the **Operations** portion of the Management Portal home page.

You can find the descriptions and explanations of these tasks in various places in the Caché documentation. Other chapters in this guide as well as the other guides in the *System Administration* documentation set describe many of the system operator tasks. The following table displays each major task and lists the most appropriate documentation source for detailed information on the topic.

Table 1–1: System Operation Tasks

Menu Item	Information
System Dashboard	“Monitoring Caché Using the Management Portal” chapter of the <i>Caché Monitoring Guide</i> .
Backup	Managing Caché Online Backups section in the “Backup and Restore” chapter of <i>Caché Data Integrity Guide</i> .
Databases	Maintaining Local Databases section in the “Managing Caché” chapter of this guide.
Processes	Controlling Caché Processes section in the “Managing Caché” chapter of this guide.
Locks	Monitoring Locks section in the “Monitoring Caché Using the Management Portal” chapter of the <i>Caché Monitoring Guide</i> .
Journals	“Journaling” chapter of the <i>Caché Data Integrity Guide</i> .
Shadow Servers	“Shadowing” chapter of the <i>Caché Data Integrity Guide</i> .
Mirror Monitor	Monitoring Mirrors section of the “Mirroring” chapter of the <i>Caché High Availability Guide</i> .
Task Manager	Using the Task Manager section in the “Managing Caché” chapter of this guide.

Menu Item	Information
System Logs	Monitoring Log Files section in the “Monitoring Caché Using the Management Portal” chapter of the <i>Caché Monitoring Guide</i> .
System Usage	Monitoring System Performance section in the “Monitoring Caché Using the Management Portal” chapter of the <i>Caché Monitoring Guide</i> .
License Usage	“ Managing Caché Licensing ” chapter of this guide.
CSP Sessions	“ CSP Session Management ” chapter of <i>Using Caché Server Pages (CSP)</i> .
Background Tasks	Background processes include any processes that are started via the Job command, plus any background server processes including CSP, Cache Direct, ODBC, or any of the objects bindings. See the JOB entry in the <i>Caché ObjectScript Reference</i> .
Diagnostic Reports	“ Using the Caché Diagnostic Report ” chapter of the <i>Caché Monitoring Guide</i> .

1.3.2 System Explorer

The database management tasks are divided into the categories displayed in the **Data Management** portion of the Management Portal home page. The following table displays each major subcategory and lists the most appropriate documentation source for detailed information on the topic.

Table 1–2: Database Management Tasks

Menu Item	Information
Classes	“ Caché Classes ” chapter of <i>Using Caché Objects</i>
SQL	“ Introduction to Caché SQL ” chapter of <i>Using Caché SQL</i>
Routines	“ User-Defined Code ” chapter of <i>Using Caché ObjectScript</i>
Globals	“ Global Structure ” chapter of <i>Using Caché Globals</i>
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 <i>InterSystems Class Reference</i> for more information.

1.3.3 System Administration

The system administrator tasks are divided into the categories displayed in the **System Administration** portion of the Management Portal home page. The following table displays each major subcategory and lists the most appropriate documentation source for detailed information on the topic.

Table 1–3: System Administration Tasks

Menu Item	Information
Configuration	“ Configuring Caché ” chapter of this guide
Security	“ System Management and Security ” chapter of the <i>Caché Security Administration Guide</i>
Licensing	“ Managing Caché Licensing ” chapter of this guide
Encryption	“ Database Encryption ” chapter of the <i>Caché Security Administration Guide</i>

2

Configuring Caché

A Caché configuration is composed of system configuration information, namespaces, databases, operator task configurations, network connections, and other advanced settings.

Use the management portal to set up a Caché system and view its configuration parameters. You can use the portal to adjust system settings as well as to create and modify namespaces, databases, and network connections, and to connect to the CSP Gateway to configure CSP applications.

The major configuration tasks are subdivided into categories, which are divided into subcategories. This chapter describes some of the topics; other topics have separate chapters or documents as references. The configuration tasks are:

- **System Configuration** — See the [System Configuration Tasks](#) table in this chapter.
- **Connectivity** — See the [Connectivity Tasks](#) table in this chapter.
- **Mirror Settings** — See the [Configuring Mirroring](#) section in the “Mirroring” chapter of the *Caché High Availability Guide*.
- **Database Backup** — See the [Configuring Caché Backup Settings](#) section in the “Backup and Restore” chapter of the *Caché Data Integrity Guide*.
- **CSP Gateway Management** — See the “[CSP Configuration](#)” chapter of *Using Caché Server Pages*.
- **SQL and Object Settings** — See the [SQL and Object Settings](#) section in the “Caché Additional Configuration Settings” chapter of *Caché Additional Configuration Settings Reference*.
- **Device Settings** — See the [Device Settings](#) section in the “Caché Additional Configuration Settings” chapter of *Caché Additional Configuration Settings Reference*.
- **National Language Settings** — See the [Configuring National Language Support \(NLS\)](#) section in this chapter.
- **Zen Reports** — See the “[Zen Reports](#)” section of the *Caché Additional Configuration Settings Reference*, and the [Configuring Zen Reports for PDF Output](#) section of the “Running Zen Reports” chapter of *Using Zen Reports*.
- **Additional Settings** — See the [Additional Tasks](#) table in this chapter.

System Configuration Tasks

Configure settings for this system from the **System Configuration** menu.

Table 2–1: System Configuration Tasks

Menu Item	Documentation Source
Memory and Startup	Configuring System Information section of this chapter
Namespaces	Configuring Namespaces section of this chapter

Menu Item	Documentation Source
Local Databases	Configuring Databases section of this chapter
Remote Databases	Remote Databases section of this chapter
Journal Settings	Configuring Journal Settings section of the “Journaling” chapter of the <i>Caché Data Integrity Guide</i>

Connectivity Tasks

Configure network connections with other systems from the **Connectivity** menu.

Table 2–2: Connectivity Tasks

Menu Item	Documentation Source
ECP Settings	“ Configuring Distributed Systems ” chapter of the <i>Caché Distributed Data Management Guide</i> .
Shadow Server Settings	Configuring Shadowing section of the “ Shadowing ” chapter of the <i>Caché Data Integrity Guide</i> .
SQL Gateway Settings	Creating Gateway Connections for External Sources in the “Using the Caché SQL Gateway” chapter of <i>Using Caché SQL</i> .
JDBC Gateway Settings	“Using the Caché SQL Gateway with JDBC” chapter of <i>Using Caché with JDBC</i> .
Object Gateway Settings	“ Setting Gateway Server Properties ” chapter of <i>Using the Caché Gateway for .NET</i> .
XSLT 2.0 Gateway Server	Configuring, Starting, and Stopping the XSLT 2.0 Gateway in the “Performing XSLT Transformations” chapter of <i>Using Caché XML Tools</i> .
Cluster Settings	Configuring Cluster Settings section of this chapter.

Additional Tasks

Configure additional settings from the **Additional Settings** menu. For a summary of additional configuration settings, see the *Caché Additional Configuration Settings Reference*.

Table 2–3: Additional Tasks

Menu Item	Documentation Source
Compatibility	Miscellaneous Settings of <i>Caché Additional Configuration Settings Reference</i> .
Advanced Memory	Advanced Memory Settings of <i>Caché Additional Configuration Settings Reference</i> .
Monitor	“ Monitoring Caché Using BMC PATROL ”, “ Monitoring Caché Using SNMP ”, and “ Monitoring Caché Using WMI ” appendixes of the <i>Caché Monitoring Guide</i> .
Source Control	“ Using Studio Source Control Hooks ” appendix of <i>Using Studio</i> .
Startup	Startup Settings in the <i>Caché Additional Configuration Settings Reference</i> .

Menu Item	Documentation Source
Task Manager Email Settings	Configuring Task Manager Email Settings in this chapter.

Most configuration changes can be done dynamically and do not require you to restart Caché. When the update does require a restart, the portal notifies you.

This chapter covers the following topics:

- [Configuring Data](#)
- [Configuring Namespaces](#)
- [Configuring Databases](#)
- [Configuring System Information](#)
- [Configuring Task Manager Email Settings](#)
- [Configuring NLS Settings](#)
- [Configuring Cluster Settings](#)

2.1 Configuring Data

Caché stores data — persistent multidimensional arrays (globals) as well as executable code (routines) — in one or more physical structures called databases. A database consists of one or more physical files stored in the local operating system. A Caché system may (and usually does) have multiple databases.

Each Caché 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 Caché. (For information about allocating the database cache, see [Memory and Startup Settings](#).)

Caché applications access data by means of a namespace. A namespace provides a logical view of data (globals and routines) stored in one or more physical databases. A Caché system may (and usually does) have multiple namespaces. Caché maps the data visible in a logical namespace to one or more physical databases. This mapping provides applications with a powerful mechanism for changing an application's physical deployment without changing application logic.

In the simplest case, there is a one-to-one correspondence between a namespace and a database, but many systems take advantage of the ability to define a namespace that provides access to data in multiple databases. For example, a system could have multiple namespaces, each of which provides a different logical view of the data stored within one or more physical databases.

For more details, see the following sections:

- [Configuring Namespaces](#)
- [Configuring Databases](#)

See the Config entries in the *InterSystems Class Reference* for information about updating namespaces, databases, and mappings programmatically.

2.2 Configuring Namespaces

A *namespace* is a collection of data and programs in a virtual work space. In a namespace, you can define the globals that various groups or people need. For example, if your accounting department needs to use certain globals that exist on different systems or in different directories, you can set up a single namespace that references all the accounting globals and databases on your network.

Caché comes with the following predefined namespaces:

- %SYS — System management information and utilities.
- DOCBOOK — Documentation.
- SAMPLES — Sample code and applications.
- USER — Empty at installation. Typically used for application development.

You can perform the following procedures for configuring 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**:

- [Create/Modify a Namespace](#)
- [Create a Namespace on an Ensemble Instance](#)
- [Rename a Namespace or Modify Default Mappings](#)
- [Add Global, Routine, and Package Mapping to a Namespace](#)
- [Delete a Namespace](#)

The size of the namespaces table is automatic and not configurable. For more information about namespaces, see the “[Namespaces and Databases](#)” chapter of the *Caché Programming Orientation Guide*.

2.2.1 Create/Modify a Namespace

You can create a new namespace at any time, but when you are first setting up the system, create the basic ones that your users need. 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.

Namespace names must be at least one character (but not more than 255 characters) long, starting with an alphabetic character or a percent sign (%) followed by an arbitrary number of alphanumeric characters, dashes, or underscores.

Important: Do not specify the following reserved system names: BIN, BROKER, DOCBOOK, DOCUMATIC, %SYS.

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 checkbox 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. After entering the required information, click **Save** to add your namespace to the configuration.

2.2.2 Create a Namespace on an Ensemble Instance

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

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

- If the default global 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 Ensemble 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 Ensemble 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 Ensemble: `/csp/namespace`

2.2.3 Rename a Namespace or Modify Default Mappings

You can rename a namespace, or change the databases to which your namespace is mapped without restarting Caché, 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 Ensemble-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 CACHETEMP; for more information, see [Using Temporary Globals and CACHETEMP](#) in the “Journaling” chapter of the *Caché Data Integrity Guide*.

5. Click **Save**.

Note: Users directly accessing the database at the time of the change may need to log out of and then back into Caché to update their namespace mapping.

2.2.4 Add Global, Routine, and Package Mapping to a Namespace

In addition to having access to the globals and routines in the mapped database, 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. For more information about mapping globals, routines, and packages, see the “[Useful Skills to Learn](#)” chapter of the *Caché Programming Orientation Guide*.

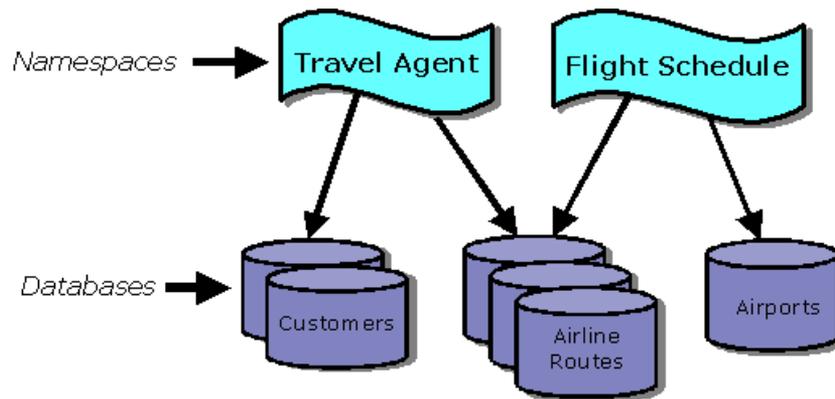
Note: Mappings are sorted alphabetically; if subscripts are specified, they are sorted by name and subscript. See the “[Global Structure](#)” chapter of the *Using Caché Globals* guide).

Click the appropriate choice to begin mapping:

- [Global Mappings](#)
- [Routine Mappings](#)
- [Package Mappings](#)
- [Mapping Data to All Namespaces](#)

The following is a schematic diagram of how mapping works in a sample airline reservation application:

Figure 2–1: Sample Namespace Mapping



Data and programs are stored in Caché *databases*, the physical storage locations, and referred to by *namespaces*, the logical references.

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.

2.2.4.1 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 Global Mapping**.
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 Global Mappings](#) in the “Global Structure” chapter of *Using Caché Globals*.

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 *Caché 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 changing 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 Terminal or Studio, by creating it on the new database and removing it from the original database.

2.2.4.2 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 Routine Mapping**.
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 [Sample 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.

2.2.4.3 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 Package Mapping**.
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 the [Package Mapping](#) section in the “Packages” chapter of *Using Caché Objects* for a description of packages and the procedure for mapping them.

For example, to make the class definitions in the Cinema package of the SAMPLES database available in the TESTSAMPLES namespace, navigate to the **Package Mappings** page and click **New Package Mapping**. Enter the information as shown in the following **Package Mapping** dialog box:

Package Mapping

A class package mapping makes all the classes within a package in a specific database visible to another namespace. Use the form below to define a new class package mapping.

Database containing the class package:

Class package to map:

Map package to namespace:

Last selected package:

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.

2.2.4.4 Mapping Data to All Namespaces

In addition to mapping globals, routines, and packages to specific namespaces, you can map them to all namespaces (except DOCBOOK and SAMPLES). To enable this form of mapping, you must first create a namespace named %ALL (see the [Create/Modify a Namespace](#) section of this guide). Then, use the procedures described in the [Add Global, Routine, and Package Mapping to a Namespace](#) section of this guide, choosing **Global Mappings > Routine Mappings** or **Package Mappings** in the %ALL namespace row.

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 (except DOCBOOK and SAMPLES).

To map data specifically to the DOCBOOK and SAMPLES namespaces, use the procedures in the [Add Global, Routine, and Package Mapping to a Namespace](#) section of this guide, choosing **Global Mappings > Routine Mappings** or **Package Mappings** in the DOCBOOK or SAMPLES namespace rows.

2.2.5 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 CSP pages from the physical path, select the check box.
3. To delete the namespace and associated mappings, click **Perform Action Now**.

2.3 Configuring Databases

A database is a CACHE.DAT file you create using the **Database Wizard**. A Caché database holds data in multidimensional arrays called *globals* and executable content called *routines*, as well as class and table definitions. Globals and routines encompass such things as methods, classes, web pages (CSP and HTML), SQL, BASIC, and JavaScript files.

CAUTION: On Windows systems, *do not* use file compression on Caché CACHE.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 CACHE.DAT file, the instance to which it belongs will fail to start, with misleading errors.

Caché 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.

You can make most database configuration changes dynamically; you can create and delete databases and modify database attributes while the system is running.

Before configuring databases on your instance, review the database considerations discussed in the next section.

Issues to consider before configuring databases in your instance and the Caché wizards for local and remote database creation are described in the following sections:

- [Database Considerations](#)
- [Local Databases](#)
- [Remote Databases](#)

2.3.1 Database Considerations

This section discusses the following topics:

- [Database Configuration Limits](#)
- [Database Configuration Considerations](#)
- [Large Block Size Considerations](#)
- [Database Compatibility Considerations](#)

2.3.1.1 Database Configuration Limits

The absolute limit on the number of databases that can be configured within a single Caché instance (given sufficient storage space) is 15,998. However, because database directory information for all databases in an instance is limited to 64KB, the practical maximum depends on the number of bytes used in their database directory paths, and is likely to be much lower. Because mirrored databases (see the [Mirroring](#) chapter of the *Caché High Availability Guide*) require additional database directory information, the precise minimum depends on the number of mirrored databases involved, if any. The different calculations for [non-mirrored](#) and [mirrored](#) databases are described in the following sections.

The number of databases that can be in use simultaneously is further limited by the operating system's limit on the number of open files (either per process or system-wide), minus what Caché reserves for its own use and devices, which is approximately half.

Calculating the Maximum Non-mirrored Databases per Instance

For non-mirrored databases, the maximum databases per instance can be calculated using the following formula:

$$\text{maximum_DBs} = 65512 / (\text{avg_DB_path_length} + 3)$$

For example, if all database directory paths are of the form `c:\InterSystems\Cache\mgr\DB\NNM`, the average length is 33 bytes, and the maximum number of databases is therefore $65512 / 36$, or 1,819.

Calculating the Maximum Mirrored Databases per Instance

A mirrored database has both a local database name and a mirror database name, which is its name within the mirror (see [Adding Databases to a Mirror](#) in the “Mirroring” chapter of the *Caché High Availability Guide*), and is referenced within ECP by its mirror database name in the format `:mirror:mirror_name:mirror_DB_name` (see [Configuring ECP Connections to a Mirror](#) in the “Mirroring” chapter). Because its mirror database name and mirror name are therefore stored along with its database directory path, the average length of the information for a mirrored database is greater. In addition, each mirrored database counts twice against the absolute maximum of 15,998. The maximum number of configurable databases on a mirror member is therefore lower and requires a more complex calculation.

The formula for the practical maximum on a mirror member depends not only on the average database directory path length but also on the mirror name length, the average mirror database name length, and the proportion of the total databases that are mirrored, as follows:

$$\text{maximum_DBs} = 65512 / ((\text{avg_DB_path_length} + 3) + ((\text{mirror_name} + \text{avg_mirror_DB_name} + 49) * \text{mirrored_DB_}\%))$$

For example, taking the database directory path of 33 bytes from the preceding example and adding a mirror name of MYMIR, a standard mirror database name of DBNNNN, and a mirrored database proportion of 80%, the maximum would be $65512 / (36 + ((5 + 6 + 49) * .8)) = 65512 / 84$, or 779.

2.3.1.2 Database Configuration Considerations

The following are tips to consider when configuring databases:

- Caché provides a seamless option to spread data across multiple physical database (CACHE.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 CACHE.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.

2.3.1.3 Large Block Size Considerations

In addition to the 8 KB (default) block size supported by Caché (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 [DBSizesAllowed](#) setting on the **Startup Settings** page (**System Administration > Additional Settings > Startup**), described in [Startup Settings](#) in the *Caché Additional Configuration Settings Reference*.
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 Caché.
4. Create the database as described in [Create Local Databases](#) in this chapter.

2.3.1.4 Database Compatibility Considerations

As described in the [Create a Local Database](#) procedure, you can copy or move a Caché database to an instance other than the one in which it was created by copying or moving its CACHE.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 the “[Backup and Restore](#)” chapter of the *Caché Data Integrity Guide*) 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 [Caché Character Width](#) in the *Caché Installation Guide*) and the same locale (see [Using the NLS Pages of the Management Portal](#) in this chapter) as the source instance (the one in which the database was created).

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 (see “Platform Endianness” in the “Supported Technologies” chapter of the online *InterSystems Supported Platforms* document for this release), 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 CACHE.DAT created on a system of different endianness, use the `cvendian` utility (see the [Using cvendian to Convert Between Big-endian and Little-endian Systems](#) section of the “Migration and Conversion Utilities” chapter of *Caché Specialized System Tools and Utilities*) 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](#) in the “Backup and Restore” chapter of the *Caché Data Integrity Guide*.

2.3.2 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](#) in the “Mirroring” chapter of the *Caché High Availability Guide*.

- **Directory** — Location of the CACHE.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](#) in the “Managing Caché ” chapter of this guide.
- **Resource Name** — The name of the database resource that governs access to the database; for more information, see the “[Assets and Resources](#)” chapter in the *Caché System Security Guide*.
- **Encrypted** — Specifies whether or not the database is encrypted; for more information, see the “[Managed Key Encryption](#)” chapter in the *Caché System Security Guide*.
- **Journal** Specifies whether or not the database is journaled; for more information, see the “[Journaling](#)” chapter of the *Caché Data Integrity Guide*.

You can use this page to:

- [Create a local database](#)
- [Edit a local database’s properties](#)
- [Delete a local database](#)

2.3.2.1 Create a Local Database

To create a local database, navigate to the **Local Databases** page (**System Administration > Configuration > System Configuration > Local Databases**).

1. Click **Create New Database** to open the **Database Wizard**.
2. Enter a database name in the text box. A database name must
 - not already be in use within the Caché instance
 - be between one and 30 characters long
 - start with an alphabetic character or an underscore; the remainder can include alphanumeric characters, dashes, or underscores
3. The first time you create a local database in a Caché instance using a particular browser, you must either
 - enter the name of the database directory, in which case this directory, containing the CACHE.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 CACHE.DAT file is created in that directory

Thereafter, by default a directory of the same name as the database name you provide, containing the CACHE.DAT file, will be 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.

4. Click **Next** to continue configuring the database. If a CACHE.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 CACHE.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.
5. 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.
 6. 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](#)” in the [Configuring Databases](#) section of this chapter.
 7. Select whether or not you want to journal globals in this database from the **Journal globals?** drop-down list. See the “[Journaling](#)” chapter of the *Caché Data Integrity Guide*.

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 CACHETEMP; for more information, see [Using Temporary Globals and CACHETEMP](#) in the “[Journaling](#)” chapter of the *Caché Data Integrity Guide*.
 8. If encryption is activated, you can encrypt this database by selecting **Yes** for **Encrypt Database?**.
 9. 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](#) in the “[Mirroring](#)” chapter of the *Caché High Availability Guide* for information about creating mirrored databases.
 10. From this panel onward, you can click **Next**. to continue configuring the database or **Finish** to accept the remaining defaults
 11. 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*)
 12. Click **Next** to view a list of the database attributes.
 13. 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 CACHE.DAT or cache.ext, even if it is not a mounted database.

2.3.2.2 Edit a Local Database's Properties

The information displayed varies depending on whether or not the database is mirrored. This section identifies the fields for:

- [Edit Non-Mirrored Local Database Properties](#)

- [Edit Mirrored Local Database Properties](#)

Edit Non-Mirrored Local Database Properties

Click **Edit** in the row of a non-mirrored database to view the following database properties and change some of them. (The “[Create a Local Database](#)” section describes many of these fields.)

- **Name**
- **Directory** (this setting must always reflect the location of the CACHE.DAT database file)
- **Encrypted** (cannot be changed)
- **Mirrored** — Click the **Add to Mirror** *mirror_name* link to add the database to the mirror in which the Caché 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](#) in the “Mirroring” chapter of the *Caché High Availability Guide* for more information.
- **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.
 - **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.
- **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 the “[Journaling](#)” chapter of the *Caché Data Integrity Guide*.
- **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 Caché starts up; if the database cannot be mounted, Caché 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 the “[Journaling](#)” chapter of the *Data Integrity Guide*). Clear to let Caché start without first mounting the database.

Note: By default, this setting is selected for required Caché databases (for example, CACHESYS, CACHELIB, CACHETEMP, and CACHE) and cannot be changed. The default is cleared for but can be selected for databases that you create, as well as some Caché databases (for example, USER and SAMPLES). For additional information about database status and explicitly dismounting and mounting databases, see The Local Databases List Information table in [Maintaining Local Databases](#) in the “Managing Caché” chapter of this guide.
- **Mount Clustered at Startup** — Select this check box to mount the cluster databases at startup.

Note: This prompt displays only if this is a clustered system.

- **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.

Edit Mirrored Local Database Properties

Click **Edit** in the row of a mirrored database to view and change some of the following database properties; see definitions in the previous section.

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 CACHE.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: 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 Caché starts up or, if a mirror member, becomes primary; if the database cannot be mounted, Caché 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 the “[Journaling](#)” chapter of the *Data Integrity Guide*) and that open transactions on the former primary have been rolled back as part of failover. Clear to let Caché 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](#) in the “Managing Caché” chapter of this guide.
- **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.

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](#) in the “Mirroring” chapter of the *Caché High Availability Guide*.

2.3.2.3 Delete 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 CACHE.DAT file, *if and only if*:
 - No other databases use this CACHE.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 CACHE.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 CACHE.DAT file, the database is still removed from the [Databases section](#) of the Caché parameters file and therefore from the list of local databases displayed by the management portal.

2.3.3 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.

From the **Remote Databases** page you can perform the following tasks:

- [Add a remote database](#)
- Edit a remote database’s properties
- [Delete a remote database](#)

2.3.3.1 Add a Remote Database

You can define a remote database on the local server if the database’s host is configured on that server as an ECP remote data server. To configure a remote data server:

1. Navigate to the **ECP Settings** page (**System Administration > Configuration > Connectivity > ECP Settings**).
2. Click **Add Remote Data Server** and enter the following information for the ECP remote data server:

- a. **Server Name** — Enter a logical local name for the remote data server for convenience of the application system administrator.
 - b. **Host DNS Name or IP Address** — Specify the host name either as a raw IP address (in dotted-decimal format or, if IPv6 is enabled, in colon-separated format) or as the Domain Name System (DNS) name of the remote host. If you use the DNS name, it resolves to an actual IP address each time the application server initiates a connection to that ECP data server host. For more information about IPv6 addressing, see the [IPv6 Support](#) section in this chapter.
 - c. **IP Port** — The port number defaults to 1972; change it as necessary to the superserver port of the Caché instance on the remote server.
3. Click **Save**.
 4. In the list of remote servers, verify the status is *Normal*. If it is not, click **Change Status** and change the status to *Normal*.

To add a remote database, follow these steps:

1. Navigate to the **Remote Databases** page (**System Administration** > **Configuration** > **System Configuration** > **Remote Databases**) and click **Add Remote Database** to launch the wizard.
2. Select **Select databases from a list** to let the portal provide you with a drop-down list of remote data servers and then a drop-down list of database directories on the server you select. If a remote data server cannot currently be reached, its database directories are not available for selection.
3. Select **Enter your own database specification** to enter the remote data server name and database directory directly. (Note that the portal does not validate your entries.)
4. Enter a database name (its name on the local server; it does not need to match its name on the remote server). You have defined a remote database.

Database names are between 1 and 30 characters long, can start with an alphabetic character or an underscore. The remaining characters can be alphanumeric, a dash, or an underscore.

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 Caché 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 view and change the database described in the preceding procedure.

2.3.3.2 Delete 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.

2.4 Configuring System Information

Caché stores system-wide configuration information in one or more configuration files in the installation directory with filenames ending in the .cpf extension. In most cases, the default configuration file created during installation, cache.cpf, is the only one you need to use Caché, but you can maintain different versions and specify a particular version when you start Caché. All but a few of the settings in the current configuration file can be modified using the management portal, but you can also modify the settings in a configuration file using a text editor. See the [Caché Parameter File Reference](#) for more information on the .cpf file.

There are some [startup settings](#) that you must change following installation, and others you should review. There are also a variety of advanced options available; however, these topics are not critical to running most Caché systems. These advanced options are described in various Caché topic-specific guides and reference books that you can access from the documentation home page.

2.4.1 Memory and Startup Settings

When you first install Caché, you may change some default system information. The **Memory and Startup** page (**System Administration > Configuration > System Configuration > Memory and Startup**) lets you allocate memory to routine and database caches and change a few startup settings.

Allocating memory for these caches is one of two primary actions you must take in determining the way a Caché instance uses memory. The other important component of Caché memory configuration and allocation is the generic memory heap (also known as the shared memory heap), which determines the memory available to Caché for purposes other than the routine and database caches. See [gmheap](#) in the “Advanced Memory Settings” section of the *Caché Additional Configuration Settings Reference* and [Generic \(Shared\) Memory Heap Usage](#) in the “Monitoring Caché Using the Management Portal” chapter of the *Caché Monitoring Guide* for more information about the **gmheap** setting.

For an in-depth look at Caché memory planning by an InterSystems senior technology architect, see [InterSystems Data Platforms and Performance Part 4 - Looking at Memory](#) on InterSystems Developer Community.

Important: When Caché is first installed, routine and database cache memory allocation is set to **Automatically**, under which Caché allocates a conservative fraction of the available physical memory for the database cache (global buffers), not to exceed 1 GB. *This setting is not appropriate for production use.*

1. Before deploying the system for production use or performing any tests or benchmarking intended to simulate production use, you must manually create an appropriate memory allocation for database cache (typically as much memory as possible after taking into account the needs of application and operating system processes) by selecting **Manually** and specifying allocations as follows.

- **Memory Allocated for Routine Cache (MB)** — Specifies the system memory allocated for caching server code. Caché assigns half the total space allocated for routines to a pool of 64 KB buffers, three-eighths of the space for a pool of 16 KB buffers, and one-eighth of the space for a pool of 4 KB buffers.

The maximum number of buffers allocated to any pool is limited to 65,529. Caché also never allocates fewer than 205 buffers to any sized pool. This means the actual memory used for routine buffers can be larger than specified in the configuration file. The format for Caché routines does not allow more than 32,768 characters for literal strings regardless of the setting for the maximum routine size.

See [routines](#) in the “[Config]” section of the *Caché Parameter File Reference* for information about allocating memory for routine buffers using the Caché parameter file (cache.cpf).

- **Memory Allocated for [blocksize] Database Cache (MB)** — Specifies the system memory allocated for buffering data, that is, for creating global buffers. You must enter a separate allocation for each database block size enabled by the **DBSizesAllowed** setting on the **Startup Settings** page (**System Administration > Additional Settings > Startup**);

see [DBSizesAllowed](#) in the *Caché Additional Configuration Settings Reference* for more information. For important information about selecting the appropriate block sizes for your applications, see [Large Block Size Considerations](#).

The number of global buffers created by this allocation for each allowed database block size is determined by dividing the allocation by the database block size; the smaller the block size, the larger the number of global buffers for databases with that block size. (This is why the database cache and the memory allocated to it are sometimes referred to as the *global buffer pool*.)

Important: If you are configuring an ECP data server, see [Memory Use on Large ECP Systems](#) in the “Developing Distributed Applications” chapter of the *Caché Distributed Data Management Guide* for important information about adjustments to database cache sizes that may be necessary.

2. You can change the **Maximum per Process Memory (KB)** allocation (that is, the maximum memory allocation for a process) for this Caché instance. The default is 262144 KB; the allowed range is 128 KB to 2147483647 KB.

Note: It is not necessary to reset this value unless you have set it lower than its default (262144 KB). If you receive <STORE> errors, you should increase the size.

This amount of 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 maximum is reached. The initial allocation is 128 KB. Once this memory is allocated to the process, it is not deallocated until the process exits.

3. On Windows platforms, you can set your Caché instance to start automatically when the system starts by selecting the **Auto-start on System Boot** check box.

Note: The **Auto-start on System Boot** check box is selected by default. If you do not want the instance of Caché to start automatically on system boot, clear the check box.

4. If you select the **Enable Long Strings** check box, Caché allocates a large string stack to handle long strings for each process.
5. You can change the **Superserver Port Number** (TCP port used to accept incoming client requests) for this Caché instance. When you change it, a `restart required` message will be displayed, indicating that the change will not take effect until you restart this Caché instance.
6. You can select a predefined label to displayed in the title bar from the **System Mode** drop-down list.
7. Click **Save** to save your modifications; restart Caché to activate them.

Some changes on this page require a Caché restart and some do not. If you modify a field that requires a restart, no changes — even those that normally do not require a restart — to your configuration take effect until you restart Caché.

Important: If you have made changes system-wide to the configuration settings that require a Caché restart, you receive the following:

```
Modification saved. You must restart system for the new values to take
effect.
```

After you close the page, the warning message does not appear again to remind you that a restart is required.

2.4.2 IPv6 Support

You can enable or disable the use of IPv6 addresses in Caché by navigating to the **Startup Settings** page (**System Administration > Additional Settings > Startup**) page; in the **IPv6** row, click **Edit**, then enter **1** to enable or **0** to disable.

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

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

Important: If Caché is running in an IPv6 or mixed network, the license server must be configured on a host running Caché 2009.1 or later; license servers running in Caché 5.1 through Caché 2008.2 do not accept IPv6 connections. See the [Configure License Servers](#) section in the “Managing Caché Licensing” chapter of the *Caché System Administration Guide*.

Caché 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 a Caché 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 Caché 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)
- 0:0:0:0:0:0:1 (full IPV6)
- ::1 (compressed IPv6)

as valid representations of the loopback address.

Generally, when asked for an Internet address that has been supplied to a Caché service earlier, Caché 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: Caché does not support the use of wildcard characters or ranges in IPv6 addresses.

2.5 Configuring Task Manager Email Settings

On the **Task Manager Email Settings** page (**System Administration > Configuration > Additional Settings > Task Manager Email**), you can configure the settings the Task Manager uses for the email notifications described in the [Using the Task](#)

[Manager](#) section of the “Managing Caché” chapter of this guide. For information about the settings, see “[Task Manager Email Settings](#)” in the *Caché Additional Configuration Settings Reference*.

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

2.6 Configuring National Language Support (NLS)

A national language locale defines the character set in which all textual data is encoded by Caché. The character set can be the 16-bit Unicode UCS-16 or one of a number of 8-bit character sets in common use worldwide, such as ISO 8859-1 (also known as Latin-1) or Windows Code Page 1252. A given national language may have several locales associated with it, at the least to provide both an 8-bit (ending in **8**) and a Unicode (ending in **w**) version of the language; the appropriate locales are installed with Caché depending on whether it is an 8-bit or a Unicode instance. For example, Danish-language locales installed with 8-bit Caché include **dai8** (Latin-9), **dan8** (Latin-1), and **daw8** (Windows-1252), while a Unicode instance has **danw**.

Each locale contains a number of character tables used by Caché when displaying text, collating data (see the “[Collation](#)” chapter of *Using Caché SQL*), 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 Caché 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 Caché change. While you can you import a locale of the wrong character width, you cannot install it as the current locale; for example, you can import **dan8** into a Unicode instance, but cannot install it as the current locale.

Installing a new locale does not result in any data conversion, but rather changes how data is represented. Because Unicode data can be accessed from any Unicode locale, changing one Unicode locale to another does not cause any data incompatibility. However, moving from one 8-bit locale to another may cause data incompatibility, depending on the character sets involved and the data stored in the system. For example, data encoded in ISO 8859-2 (Latin-2) will be interpreted differently by a locale based on CP1251 (Windows Cyrillic). The only guarantee is that the lower part of all 8-bit character sets (characters 0-127) is equal to the ASCII character set.

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 Caché, which are overwritten when the instance is upgraded.

2.6.1 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](#)

2.6.1.1 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 Caché. When writing ObjectScript code or using some utilities, it is possible to specify a particular table for a given purpose; the default table is used 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](#).in the “Customizing the Caché System” chapter of *Caché Specialized System Tools and Utilities*.

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

2.6.1.2 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 Caché. 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.

2.6.1.3 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 **OK**. The only valid file extensions are `.xml` and `.goq`.
3. A message displays indicating how many locales, tables, and subtables have been imported.

2.6.2 Using the NLS Class Packages

The [System Classes for National Language Support](#) section of the “Customizing the Caché System” chapter of *Caché Specialized System Tools and Utilities* 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*.

2.7 Configuring Cluster Settings

These settings apply only to platforms that support clusters. Configuring them does not automatically force a system to join a cluster. A system joins a cluster automatically the first time it mounts a database for clustered access.

The connection is automatically configured and the cluster members do not need to be listed as clients of each other. The only requirement is that if the machine has multiple IP addresses (generally because there are multiple network interface cards) you must set the **CommIPAddress** to force Caché to use a specific IP address for the cluster ECP traffic.

The following settings appear in the Cluster Settings category on the **Cluster Settings** page (**System Administration > Configuration > Connectivity > Cluster Settings**):

- **JoinCluster** — True or false. When true, this configuration is part of a cluster and the appropriate cluster definition settings can be configured; in addition, the directory must exist. The default is false.
- **CommIPAddress** — This setting is required only for systems with multiple interfaces to specify the default IP Address for other cluster members to use for inter-cluster communication; it is advertised in the PIJ. The string, which may be up to 128 characters long (a cluster feature), is a resolvable DNS name or IP address.
- **pijdir** — The pre-image journal (PIJ) file to support cluster failover, recovery, and write image journaling in a cluster. This setting must be the same on each cluster node and is required when **JoinCluster** is set to true. The directory must exist.

If you edit this setting, you must restart Caché to apply the change.

3

Managing Caché

This chapter explains common Caché operations tasks including displaying process details, broadcasting messages, and monitoring processes. This chapter discusses the following topics:

- [Maintaining Local Databases](#)
- [Controlling Caché Processes](#)
- [Using the Task Manager](#)
- [Using the Background Tasks Page.](#)

3.1 Maintaining Local Databases

You can review and maintain local databases on the Databases page of the management portal, which you can reach by selecting **Systems Operations** and then **Databases**. For each local database, you see the following information:

Table 3–1: Local Databases List Information

Column Heading	Definition
Name	The database name; click this name to display more details about this database.
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.

Column Heading	Definition
Status	<p>The status of the database: mounted (including which permissions it has), unmounted, or dismounted.</p> <ul style="list-style-type: none"> A mounted database is one for which Mount Required at Startup is selected, as described in Edit Database Properties in the “Configuring Caché” chapter of this guide, and which therefore must be mounted for Caché to start or become primary in a mirror; in this case, it is always mounted and accessible when Caché 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 Caché. An unmounted database is one for which Mount Required at Startup is not selected and which therefore does not need to be mounted for Caché 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 Caché. A dismounted database is one that has been explicitly dismounted; it is inaccessible until you explicitly mount it or restart/stop Caché (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 in the “Configuring Caché” chapter of this guide. <p>Note: For information about explicitly mounting or dismounting a database, see the Dismount / Mount action buttons in this table.</p>
Encrypted	Indicates whether or not the database is encrypted.
Journal	Indicates whether globals in the database are journaled with a Y or an N.
Dismount / Mount	Buttons that let you explicitly dismount or mount a database. The new status remains in effect until you explicitly change it or restart/stop Caché; to permanently dismount a database, you must remove it from the configuration.

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 `cac*` 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).

3.1.1 Displaying Free Space Information

To get free space information, you can display the [Database Freespace page](#) 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.

3.1.1.1 Display Free Space Information Using the Management Portal

To display the Database Freespace page, which shows information about the amount of free space on each local database and lets you [compact](#) and/or [truncate](#) databases, navigate to the Databases page and click **Freespace**. The following table describes the information displayed on the Database Freespace page:

Table 3–2: Local Databases Freespace Information

Column Heading	Definition
Name	The database name.
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.
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.

Managing the free space (empty blocks) in a database is an important aspect of database maintenance. You can perform the following free space management operations on the Database Freespace page

- [Compact a database](#) — Move free space distributed throughout a database to its end
- [Truncate a database](#) — Return free space at the end of a database to the underlying file system

In addition, you can use the `^DATABASE` utility for the following database management operations involving global blocks (blocks containing global data):

- [Compact globals in a database](#) — Reduce unused space within global blocks and the overall number of global blocks, increasing free space
- [Defragment globals in a database](#) — Relocate global blocks so that the blocks representing a given global are in continuous sequence

The `^DATABASE` utility can also be used to display free space information and to compact and truncate databases, along with performing other database tasks such as recreating a database, which lets you clear the data in an existing database without changing the database’s name or size. See `^DATABASE` in the “Using Character-based Security Management Routines” chapter of the *Caché Security Administration Guide* for information about the utility.

Note: The data structures used by Caché 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 indices to regain performance.

3.1.1.2 Display Free Space Information Using `^%FREECNT`

Caché 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:\MyCache\mgr\
c:\MyCache\mgr\cacheaudit\
c:\MyCache\mgr\cachelib\
c:\MyCache\mgr\cache\
c:\MyCache\mgr\cachetemp\
c:\MyCache\mgr\docbook\
c:\MyCache\mgr\samples\
c:\MyCache\mgr\user\
Device:
Right margin: 80 =>
```

```
Cache Database Free Space
Feb 15 2012 7:25 PM
Database Max Size Size Available %Free Disk Free
c:\MyCache\mgr\ Unlimited 191MB 19MB 9.94 60.79GB
c:\MyCache\mgr\cacheaudit\ Unlimited 1MB 0.43MB 43 60.79GB
c:\MyCache\mgr\cachelib\ Unlimited 319MB 27MB 8.46 60.79GB
c:\MyCache\mgr\cache\ Unlimited 1MB 0.55MB 55 60.79GB
c:\MyCache\mgr\cachetemp\ Unlimited 4MB 1.5MB 37.5 60.79GB
c:\MyCache\mgr\docbook\ Unlimited 144MB 11MB 7.63 60.79GB
c:\MyCache\mgr\samples\ Unlimited 114MB 52MB 45.61 60.79GB
c:\MyCache\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 ^%FRECNT
```

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

```
Cache Database Free Space
Feb 15 2012 7:28 PM
Database Max Size Size Available %Free Disk Free
c:\MyCache\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 filesystem to handle database expansion.

You may choose the device to which to send the information and choose the line length of the display.

3.1.2 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 moves enough global blocks from the end to the beginning 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**) and click **Freespace** to display the [Database Freespace page](#).

2. Click **Compact** in the row of the database you want to compact. This displays the **Compact Database** page, 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.

Note: If a database is mounted read only and the Caché instance is not a backup or async mirror member (see the “[Mirroring](#)” chapter of the *Caché High Availability Guide*), the **Compact** link does not appear in its row on the **Database Freespace** page.

3. The **Target freespace (in MB) at end of file** prompt allows you to specify the amount of free space you want at the end of the file following the operation. 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.

4. When the Compact dialog box displays a message that the background job has started, click the **Click here to view the background tasks page** link near the top of the page to view the status of Compact Database Space background tasks on the **Background Tasks** page (**System Operation > Background Tasks**).
5. In the Compact dialog box, click **Done** to redisplay the Database Freespace page. If **>>** is displayed in the first column of the database row, refresh the page to see the new size displayed in the **Size** column.

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.

3.1.3 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 [compact](#)ed, 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 or cluster-mounted volumes.

To truncate a database:

1. Navigate to the **Databases** page (**System Operation > Databases**) and click **Freespace** to display the [Database Freespace page](#).

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.

2. Click **Truncate** in the row of the database you want to truncate. This displays the **Truncate Database** page (**System Operation > Databases > Freespace > Truncate**), which shows the name, location, and current size in megabytes (MB) of the selected database.

Note: If a database is mounted read only and the Caché instance is not a backup or async mirror member (see the “[Mirroring](#)” chapter of the *Caché High Availability Guide*), the **Truncate** link does not appear in its row on the **Database Freespace** page.

3. 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.
4. When the page displays a message that the background job has started, click the **Click here to view the background tasks page** link near the top of the page to view the status of background tasks on the **Background Tasks** page (**System Operation > Background Tasks**).
5. In the Truncate dialog box, click **Done** to redisplay the Database Freespace page. If **>>** is displayed in the first column of the database row, refresh the page to see the new size displayed in the **Size** column.

3.1.4 Compacting Globals in a Database

The **Compact globals in a database** option of the `^DATABASE` routine consolidates global data into fewer blocks, increasing the amount of free space in a database.

When globals are created and updated, Caché 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 Caché 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](#) in the chapter “Introduction to Data Integrity” in the *Caché Data Integrity Guide*) 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 for a numbered list, and then a list of numbers, for example `1,4,7-10`.
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, indicate whether you want to display the results of the compacting operation for each global, specify the output device, 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](#) in the “Configuring Caché” chapter of this guide), or if expansion is not possible because there is insufficient space available on the storage volume, the operation is canceled.

3.1.5 Defragmenting Globals in a Database

The **Defragment globals in a database** option of the `^DATABASE` routine 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 globals in 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 CACHETEMP 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 so you can first [compact](#) the database and then choose the **Defragment globals in a database** option again, reducing the amount of expansion required to complete the defragmentation operation..

To defragment 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 **14) Defragment globals in a database** from the menu.
3. Specify the directory of the database on which you want to run the operation.
4. Indicate that want to allow expansion, if required, and complete.

Note: The defragment 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 [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](#) in the “Configuring Caché” chapter of this guide), or if expansion is not possible because there is insufficient space available on the storage volume, the operation is canceled.

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

3.2 Controlling Caché Processes

A Caché system runs a number of processes. Application code as well as Caché system code executes within these processes. There are three categories of Caché processes:

- User processes, created when a user connects to Caché.
- 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](#)).
- Caché system processes.

In this chapter, the word “process” by itself refers to both user and background processes.

You can manage and control processes using the Management Portal:

Table 3–3: 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.
Manage process refresh interval	Display the Processes page; optionally, click Details in the right hand column to display the Process Details page. In either case, edit the Refresh options (on the operations bar), as desired.

3.2.1 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 3–4: Process Column Information

Column Heading	Definition
Job #	Index of the Processes table.
Process ID	Operating system process identification number (PID).*
User	Name of the user who owns the process.
Device	Current device the process is using.
Namespace	Namespace in which the process is running.
Routine	Name of the routine that the process is currently executing.
Commands	Number of commands executed.

Column Heading	Definition
Globals	Number of global references, including updates, executed (database reads and writes) since the process entered Caché.
State	Process state. See the <i>State</i> property of the %SYS.ProcessQuery class documentation in the <i>InterSystems Class Reference</i> 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 Caché in programmer mode. A plus or minus sign appears next to Callin processes:

- + Process is in Caché
- – Process is not in Caché

The [Callin API](#) is a Caché facility that lets you execute and evaluate ObjectScript commands and expressions from within C programs.

3.2.2 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. The portal displays the **Process Details** page for the process you selected.

Each category is described in one of the tables that follow:

- [General information](#)
- [Client application details](#)
- [Execution details](#)

In most cases, you can click any of the indicators to display a description of the item in the bottom detail box at the lower left corner of the page of the **Process Details** page.

Table 3–5: General Information

Indicator	Definition
Process ID	Process ID (PID) number of this process.
User Name	Name of the user currently logged in for this process.

Indicator	Definition
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.
Commands Executed	Number of commands executed by this process.
Memory Limit	Amount of memory (Kbytes) allocated for use by this process.
Memory Peak	Peak amount of memory (Kbytes) used by this process.
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.
Memory Used	Amount of memory (Kbytes) 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 3–6: Client Application Details

Indicator	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 3–7: Execution Details

Indicator	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.
Routine	Name of the routine this process is currently executing.
Source Location	Last reported source location (routine name plus offset) of this process.
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.

3.2.2.1 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.

3.2.2.2 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.

3.2.2.3 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.

3.2.3 Broadcast Messages to the 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 Caché 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**.

3.2.4 Manage Refresh Interval

You can control whether or not to auto-refresh the process information and, if you want to refresh the information, you can specify how frequently it is refreshed. The minimum interval for auto-refreshing the data is 5 seconds; the default interval is 10 seconds.

Note: Auto-refresh is turned off by default; to refresh processes manually, click the refresh icon.

To turn on auto-refresh for all active processes, do the following in the Management Portal:

1. Display the **Processes** page (**System Operation > Processes**).
2. Click the **on** (on the options bar) and enter the interval (minimum: 5 seconds) in the text box.
3. To turn on auto-refresh for process details, do the following in the Management Portal:
 - a. Display the **Processes** page (**System Operation > Processes**).
 - b. 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.
 - c. Click the **on** (on the options bar) and enter the interval (minimum: 5 seconds) in the text box.

3.3 Using the Task Manager

To use the Task Manager, select **System Operation > Task Manager** to display the following menu options:

- [New Task](#) (schedule a new task)
- [On-demand Task](#)(execute a task on demand)
- [Upcoming Tasks](#) (view tasks scheduled to run in the next 24 hours)
- [Task Schedule](#) (view all scheduled tasks)
- [Task History](#) (view completed tasks)
- [Import Tasks](#) (import tasks previously exported to files, possibly by another Caché instance)

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.

3.3.1 New Task

The **New Task** option starts the **Task Scheduler Wizard**.

1. The first page of the wizard asks for the following information:

- **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 (the list varies with the namespace you select at the previous prompt)

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).

- `CheckLogging` — creates an alert or turns off `^%ISCLLOG` or `^ISCSOAP` logging if it has been running too long (by default, 2 days)
- `CleanSQLIndex` — cleans up stale SQL Statement Index entries
- `CumuIncrDBList` — a cumulative backup of databases in the defined list
- `DiagnosticReport` — collects and delivers diagnostic reports to the WRC
- `FullAllDatabases` — a full backup of all databases
- `FullDBList` — a full backup of databases in the defined list
- `IncrementalDBList` — an incremental backup of databases in the defined list
- `IntegrityCheck` — an integrity check of databases in the namespace. The Integrity Check task only appears when the `%SYS` namespace is selected.
- `InventoryScan` — compiles an inventory scan of the instance
- `PurgeAudit` — purge the audit database after a specified time following a journal switch
- `PurgeBackupLog` — purge the backup log after a specified time following a database backup
- `PurgeErrorsAndLogs` — purge error globals and `cconsole.log` based on settings in the `cache.cpf` file (see [ErrorPurge](#) and [MaxConsoleLogSize](#) parameters in the Startup section of the “Introduction to the Caché Parameter File” chapter of the *Caché Parameter File Reference*)

Note: When [System Monitor](#) is running, it monitors and limits the size of the console log.

- `PurgeJournal` — purge journal files that meet the purging criteria
- `PurgeTaskHistory` — purge task manager history files
- `RunLegacyTask` — run a legacy task, that is, 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. Caché 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` — disables expired user accounts, and expires user passwords and CSP session tokens

- `ShadowPurge` — purge shadow files
- `SuspendOnError` — determines what happens if the Task returns an error `%status` from the `OnTask()` method. Errors encountered by the Task Manager trying to execute the task (login error or no such Task found) do not check this flag.
- `SwitchJournal` — switch to a new journal file
- `UpdateSQLStats`— updates the information on SQL query statistics

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

- **Task priority** — Choose from **Priority Normal** > **Priority Low**, or **Priority High**.
- **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 Caché username you select; see [License Login Special Considerations](#) in the “Managing Caché Licensing” chapter of this guide for more information.

- **Open output file when task is running** and **Output file** — If the task creates output, the log file is written to the directory specified.
- **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.
- **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.
- **Send completion email notification to** — If you have configured email settings for the Task Manager (see [Configuring Task Manager Email Settings](#) in the “Configuring Caché” chapter of this guide), enter a comma-separated list of email addresses to which a notification should be sent when the task ends successfully.
- **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.
- **How should task run for Mirror** — If this instance is a mirror member, specify the type of member the task can be run on:
 - run on primary failover member only
 - run on backup failover member and async members only (all except primary)
 - run on all mirror members (primary, backup, and asyncs)

Important: This setting must be set for all tasks on the mirror member, the task will not run on a mirror member if not. Adding an instance to a mirror member does not automatically update this setting 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.

2. Click **Next** to specify when you want the task to run.
3. Click **Finish** to schedule the task.

All tasks are configured to expire by default. To change this:

1. Open the Terminal.
2. Enter `zn "%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.`

The selected task will no longer expire.

There are three additional expiration fields that control when a task expires. These are controlled by days, hours, and minutes. To adjust when a specific task expires:

1. Open the Terminal.
2. Enter `zn "%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 `Expires in how many days?, Expires in how many hours?, or Expires in how many minutes?`
7. To edit the expiration times of this task, enter the days, hour, or minutes you would like to schedule.

The values for this expiration are now saved.

For details on the information necessary to schedule or edit a task, see the %SYS.TaskSuper class documentation in the *InterSystems Class Reference*.

3.3.2 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.

3.3.3 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.

3.3.4 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:

- *taskname* — 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.

3.3.4.1 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** — Lets you change the task definition and schedule using the [Task Scheduler Wizard](#).
- **History** — Lets you view the task's [history](#).
- **Resume > Suspend** — Let you suspend a task or resume a suspended task, as on the [Upcoming Tasks](#) page.
- **Delete** — Lets you permanently delete the task.
- **Export** — Lets you export a task to a file that can later be [imported](#), including by another Caché instance.

- **Run** — Lets you 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 options described are unavailable while a task is running.

3.3.4.2 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.

3.3.5 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.

3.3.6 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.

3.4 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](#) earlier in this chapter, 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 Caché Processes](#).

4

Managing Caché Licensing

This chapter contains an overview of the Caché license system; it covers the following topics:

- [Configuring Caché Licensing](#)
- [Activating a License Key](#)
- [Determining License Capacity and Usage](#)
- [Identifying Users](#)
- [Application Licensing](#)

Important: Licenses from versions prior to Caché 5.1 do not work with this version of Caché. Please contact the [InterSystems Worldwide Response Center \(WRC\)](#) for an appropriate key if you are upgrading from an older version.

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

4.1 Configuring Caché Licensing

Each Caché instance maintains an independent local view of its license capacity and current use. Each instance requires access to the key; therefore you must install and activate a license key file (typically named `cache.key`) on every instance, except evaluation installations.

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 Caché license servers to allocate the Caché 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. License servers can run on any computer where you run a Caché instance. A monitor process sends update messages to the license server, which coordinates license allocation when more than one instance shares a license.

The license server coordinates the views of license use maintained locally in every instance. The license server is not a Caché process; it is unaffected if a Caché instance shuts down. One license server can handle multiple instances. Therefore, you need at most one per host regardless of how many Caché instances run on a host. However, each Caché instance must have a local copy of the authorizing license key file installed.

If you run Caché servers on multiple hosts, you can configure more than one license server to provide redundancy. The license software selects one of the license servers to be the active server. The other servers are available to take over should the active server fail. This is much less critical than with previous Caché releases because the instance can continue running with users logging in and out in the absence of the license server, and the license server continues running after shutdown when it is supporting more than one instance. When configuring license servers, decide which server or servers you want to host the license server. You can configure it to run on as many hosts as you want, but more than three is excessive. Since the license server is started by a running instance, it should be configured to run on systems where you expect a Caché instance to be running consistently.

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 `cache.key` file and all instances authorized by the same key use the same license servers. However license units are not summed across license keys. Cache instances using different license keys do not share license units, and users logged into two instances using different license keys will consume a separate license unit from each key.

4.1.1 Configuring License Servers

Configure the license servers using the Management Portal:

1. Navigate to the **License Servers** page (**System Administration** > **Licensing** > **License Servers**).
2. This 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. From this page you can edit or delete an existing server definition or add a new server.

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

3. Click **Add** to configure a license server.

Enter a name for the license server in the **Name** box, the IP address of the host on which it runs in the **Hostname/IP Address** box, and the port number used by the license server in the **Port** box.

The license server name identifies the license server in the configuration and must be unique to a configuration.

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:feec:ba23:cd1f:dcb1:1010:9234:4085).

The license server port number must be a number between 1024 and 65535; InterSystems uses a default port number of 4001. The port numbers of redundant license servers running on different hosts do not need to be unique, but must be different from any port number used at that IP address.

4. Click the name of a listed license server to update the information described in the previous step.
5. Click **Delete** to remove a license server from the configuration.

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.

4.2 Activating a License Key

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

After installing Caché, 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 manager directory (`install-dir/mgr`) as `cache.key`, if it is not named that already.

Note: You can also select a license key during Windows installation (see the chapter “[Installing Caché on Microsoft Windows](#)” in the Caché Installation Guide). When you do this, the license is automatically activated and the license key is copied to the instance's manager directory as `cache.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 **[System] > [Licensing] > [License Key]** page. 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 (typically but not necessarily named `cache.key`). 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 will be 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 manager directory as `cache.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 will enable fewer features than the current license if this is the case.

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 generic 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 console log. You can increase the size of the `gmheap` setting from the **[System] > [Configuration] > [Advanced Memory Settings]** page of the Management Portal.

If the new license key consumes at least 1000 64 KB pages more `gmheap` space than the existing key, the Cache instance must be restarted to fully activate the new license key. However, since each page represents 227 licenses on a unicode system and more on an 8-bit system, this situation is rarely encountered.

4.2.1 License Troubleshooting

If, after entering your license and restarting Caché, only one user can log in, use the management portal to investigate. The **[System] > [License Usage]** page shows how many processes are running when you select **By Process**. You can also use

the portal to display license information from the **[System] > [Licensing] > [License Key]** page, 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 console log and System Monitor log, which can be viewed in the portal on the **[System] > [System Logs] > [Console Log]** page and **[System] > [System Logs] > [System Monitor Log]** page, respectively (see [Monitoring Log Files](#) section of the “Monitoring Caché Using the Management Portal” chapter of the *Caché Monitoring Guide*). Caché System Monitor writes license expiration warnings and alerts to these logs, while Caché Health Monitor writes license acquisition alerts and warnings. When the license limit is exceeded, alerts are written to the console log by the licensing module. In Caché Application Monitor, you can configure license metric-based alerts to send email notifications or call notification methods. See the “Using Caché System Monitor” chapter of the *Caché Monitoring Guide* for more information about these monitoring tools.

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

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

This document describes many of these methods. If your license problem prevents you from obtaining a terminal session, open a Command Prompt window run as Administrator, change to *install-dir/bin*, and run the following command to get one additional terminal session within the Command Prompt window for license troubleshooting purposes:

```
cache -s ..\Mgr -B
```

4.2.2 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 *install-dir\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:

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

For more information on the **csession** command, see [Connecting to a Caché Instance](#) in the “Using Multiple Instances of Caché” chapter of this guide.

4.3 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 Caché license application programming interface (API) and presents a number of methods and related queries that can be used to query license capacity and current use. You can run these class methods using the special **\$System** object. See the **%SYSTEM.License** class entry in the *InterSystems Class Reference* for details.

You can also run the methods using the **[System] > [License Usage]** page in the management portal, as detailed in the following table:

Link on [System] > [License Usage]	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	Summary() — returns current distributed license usage sorted by users as displayed in the <code>Distributed Licence Use</code> section of \$System.License.ShowSummary . (Link is disabled when no license server is connected.)

You can also use the following class methods 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:

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

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

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

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

```
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 CSP Gateway connections that can be made. From the SMP in the CSP Gateway Management section:

1. Navigate to **Server Access**.
2. Select **State-less Paramaters**.
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 will slow down. That would be the indication that you need to buy more licenses.

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

- [Methods to Show Local License Information](#)
- [Methods to Show License Server Information](#)

4.3.1 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:

```
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:

```
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:

```
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

4.3.2 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 Caché instances currently supported by the license server. The `Local license use` section presents a view of license use for the single Caché instance in which the program is run:

```
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:

```
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:

```
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:

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

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

```
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 Caché 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.

4.4 Identifying Users

The Caché 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 CSP 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 Caché 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 Caché 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, Caché 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.

InterSystems expects that most applications are moving to identify their users by name, eliminating problems associated with using a default user ID based on client IP address, CSP session ID, or other connection-derived user ID.

For example, when firewall or terminal server software is used, Caché cannot differentiate among connecting users, so it falls back on the maximum-connection transition rule. Using mixed connections, such as CSP and Caché Direct, 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](#)

4.4.1 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 Caché 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, Caché 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 system 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 Caché commands, and then every 1000 Caché 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.

4.4.2 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 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, Caché uses the client IP address to identify a user to the license tracking subsystem. If you installed Caché 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 Caché 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:

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

The following example enables, then disables username licensing:

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

See the [\\$USERNAME](#) special variable entry in the *Caché ObjectScript Reference* for more information.

4.4.3 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 CSP 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 (CSP, a terminal window, and a Caché Direct connection from a Visual Basic client). In this case, you can use the `%CSP.Session` method, `%CSP.Session.Login(username, password)` to perform an explicit license login for the session.
- Anonymous SOAP requests (that is, SOAP requests that do not require Caché logins) consume a license unit for minimum of 10 seconds; however, any SOAP request that identifies the user requires a license because it is considered a “user request.” Information about implementing a SOAP session is available in [SOAP Session Management](#) in *Creating Web Services and Web Clients in Caché*.
- Caché 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** option on the **[System] > [Task Manager]** page in the Management Portal (see [Using the Task Manager](#) in the “Managing Caché” chapter of this guide) consumes a license unit, with the license user ID based on the Caché 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`, or the older `^ZSTU`, 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.

4.5 Application Licensing

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

4.5.1 Loading an Application License

Any application license can be loaded and activated by a Caché 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 Caché license, but is tracked independently by the Caché 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 Caché, 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.

4.5.2 Application Licensing API

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

- **SYSTEM.License.LoadAppLicenseFile**
Loads an application license from a file.
- **SYSTEM.License.GetAppLicense**
Returns the contents of a currently loaded application license as a string of `keyword=value` pairs.
- **SYSTEM.License.GetAppNames**
Returns the names of applications currently licensed by InterSystems, InterSystems application partners, or both.
- **SYSTEM.License.IsAppLicensed**
Indicates whether an application or feature is licensed.
- **SYSTEM.License.TakeApplicationLicense**
Consumes a license unit for an application and feature. The license user ID of the current process (or in the case of CSP, the session license ID) is the user identifier associated with the application license consumed.
- **SYSTEM.License.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.
- **SYSTEM.License:ApplicationServerLogin**
Indicates whether the number of instances using a currently loaded application license exceeds the limit specified in the key.

- **%SYSTEM.License.GetAppLicenseValue**
Retrieves the value associated with an application license feature keyword.
- **%SYSTEM.License.GetAppLicenseMin**
Retrieves the minimum number of free license units associated with an application license feature keyword, where the value associated with the keyword is numeric.
- **%SYSTEM.License.GetAppLicenseFree**
Retrieves the number of free license units associated with an application license feature keyword, where the value associated with the keyword is numeric.
- **%SYSTEM.License:ApplicationUserList**
Returns license and application license use by user ID.

5

Using Caché on Windows

Managing a Caché instance on the Microsoft Windows platform is straightforward. You can perform most tasks using the Management Portal and the Caché Launcher. You can also control a Caché instance from a command prompt. This chapter discusses the following topics:

- [Caché Launcher](#)
- [Starting Caché](#)
- [Stopping Caché](#)
- [Removing the Caché Launcher from the System Tray](#)
- [Controlling Caché from the Command Prompt](#)
- [Connecting to Caché on the Command Line](#)

This chapter refers to the Caché installation directory as *install-dir* — you can find the current default directory in the [Default Caché Installation Directory](#) section of the *Caché Installation Guide*.

CAUTION: *Do not* use Windows file compression on Caché CACHE.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 CACHE.DAT file, the instance to which it belongs will fail to start, with misleading errors.

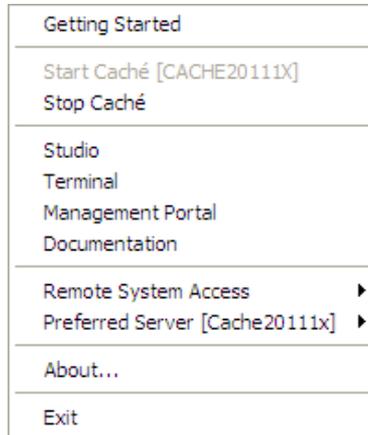
5.1 Caché Launcher

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

Correspondingly, you can initiate many of the Caché tools from the **Start** menu by pointing to **Programs, Caché**, and then to the appropriate Caché instance name.

When you start Caché on a Windows-based system, the Caché Launcher  appears in the system tray of the taskbar.

When you click the Caché Launcher, a menu appears with commands to use the Caché utilities and programming environments.

Figure 5–1: Caché Launcher Menu

The following table describes the commands available from the Caché Launcher menu.

Caché Launcher Command	Description
Getting Started	Displays links to tutorials, release notes, documentation, and other related information.
Start Caché	Starts the default instance specified in the square brackets after the menu item, for example [CACHE]. If the Caché server is already started, this option appears dimmed—it is unavailable. Note: For information about how to prevent an instance from starting automatically, see the Memory and Startup Settings section of the “Configuring Caché” chapter of this guide.
Stop Caché	Shuts down or restarts the local Caché instance. If the Caché server is stopped, this option appears dimmed—it is unavailable.
Studio	Creates, edits, deletes, and compiles Caché class definitions, CSP (Caché Server Pages) pages, Caché Basic routines, and ObjectScript routines. See Using Studio for more information.
Terminal	Invokes the command line interpreter in the Caché programming environment. See the Using the Terminal guide for more information.
Management Portal*	Performs common system management tasks. Creates databases and namespaces, and adjusts all Caché configuration settings. Displays classes, globals, and routines, and functions for managing each. Displays tables and views, perform queries and SQL management functions. See the “ Using the Management Portal ” chapter for more information.
Documentation	Displays Caché online documentation.
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 the Define a Remote Server Connection section of the “ Connecting to Remote Servers ” chapter for more information.
About	Displays Caché version and build information.

Caché Launcher Command	Description
Exit	Removes the Caché Launcher icon from the system tray; this does not stop Caché.

* Replaces the functions of Caché Explorer, SQL Manager, Control Panel, and Configuration Manager in Caché 5.0.

5.2 Starting Caché

To start Caché, 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 Caché, view the `cconsole.log` file as described in the [Monitoring Log Files](#) section of the *Caché Monitoring Guide*.

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

If the Caché Launcher is not in the system tray, from the **Start** menu point to **Programs, Caché**, the Caché instance name, and click **Start Caché**. To return the Launcher to the system tray, go to the `install-dir/Bin` directory and double-click the `csystray.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 `cache` from the `MyCache/bin` directory, enter the following command and click **OK**.

```
c:\mycache\bin\ccontrol start cache
```

These methods of starting Caché call the `ccontrol start` command. See [Controlling Caché from the Command Prompt](#) for more options and information on the `ccontrol` command.

5.3 Stopping Caché

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

From the Caché Launcher menu click **Stop Caché** to shut down or restart the local Caché instance. By default, this option shuts down (or restarts) Caché 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 **Start** menu. Point to **Programs, Caché**, the Caché instance name, and click **Stop Caché**. You cannot cancel a shutdown once the countdown reaches 0 and the shutdown procedures have started.

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

These methods of stopping Caché call the `ccontrol stop` command. See [Controlling Caché Instances](#) for more options and information on the `ccontrol` command.

Remove Stop Caché Command

To prevent unintentional execution of the **Stop Caché** command, you can remove the command from the Caché Launcher by deleting the `cstop.exe` file from the `install-dir\Bin` directory of the corresponding Caché instance. You can also remove the **Stop Caché** shortcut from the appropriate Caché instance from the **Start** menu. Point to **Programs, Caché**, the Caché instance name, then right-click **Stop Caché** and click **Delete**.

5.4 Removing the Launcher from the System Tray

You can choose to temporarily or permanently remove the Launcher from the system tray; this only affects the Launcher and not Caché as a whole.

5.4.1 Temporarily

If you click on the Launcher in the system tray and choose **Exit**, the Launcher stops and its icon no longer appear in the system tray. The Launcher reappears when the system is rebooted or you start Caché from the Start menu (see [Starting Caché](#) in this chapter).

5.4.2 Permanently

On your Windows boot disk, go to the directory,

```
Documents and Settings\All Users\Start Menu\Programs\Startup\Caché
```

Delete the directory that bears the instance name whose Launcher you wish to eliminate from the system tray.

5.5 Controlling Caché from the Command Prompt

You can control a Caché instance from the Windows command prompt by running the `ccontrol.exe` program in the `install-dir\Bin` directory. For information about the `ccontrol` command, see [Controlling Caché Instances](#) in the “Using Multiple Instances of Caché” chapter of this guide.

Note: To display the most current help file for running the `ccontrol` command, invoke `ccontrol help` from the Caché `install-dir\Bin` directory. For example:

```
C:\MyCache\Bin>ccontrol help
```

The command displays the `CcontrolHelp.html` file in your browser; the file is in the `install-dir\Help` directory.

5.6 Connecting to Caché on the Command Line

You can log into a Caché instance on the command line using the `csession` command. See [Connecting to a Caché Instance](#) in the “Using Multiple Instances of Caché” chapter of this guide for more information on `csession`.

6

Using Caché on UNIX®, Linux, and macOS

This chapter describes specific administrative procedures on UNIX®, Linux, and macOS. This chapter addresses the following topics:

- [UNIX® Users, Groups and Permissions](#)
- [Startup on UNIX®](#)
- [Managing Caché](#)

6.1 UNIX® Users, Groups and Permissions

Every Caché installation on a UNIX® platform has the following users and groups:

- *Root* — Caché must be installed by `root`, and some processing by Caché 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 Caché superserver and its jobs* — All Caché processes spawned by the superserver to serve incoming requests run as this user; in addition jobs hosted by `jobserver` processes, `taskmanager` jobs, and user-defined startup routines (for example, `^%ZSTART`) also run as this user. By default, this user is `cacheusr`, but you can change the user during installation.
- *Effective group for Caché processes* — All Caché processes automatically run as this group, which allows normal users, while inside Caché, to access Caché database and journal files to which they may not otherwise have been granted access; file permissions on these and other Caché files are set to allow this group to have appropriate access. On a secure system, no actual user should be a member of this group. By default, this group is `cacheusr`, but you can change the group during installation.
- *Group allowed to start and stop instance* — This group, `root`, and the *Owner of instance* can start and stop Caché.

All journals and journal directories must have the group ownership set to the *Effective group for Caché processes* group and grant full permissions to that group (`rwx` 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 Caché 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 Caché processes* is `cacheusr` and the *Owner of instance* is `cacheowner`, although the files may have different user ownership depending on the context in which they were created. For example:

```
journal directory    cacheowner    cacheusr     drwxrwxr-x
20170801.001        cacheowner    cacheusr     -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 Caché 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.

6.1.1 Database and Database Directory Permissions

All databases and database directories must have the group ownership set to the *Effective group for Caché 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 Caché 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 Caché processes* is `cacheusr` and the *Owner of instance* is `cacheowner`, although the files may have different user ownership depending on the context in which they were created:

```
dataset directory   cacheowner    cacheusr     drwxrwxr-x
CACHE.DAT           cacheowner    cacheusr     -rw-rw----
```

6.2 Startup on UNIX®

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

1. The `cache.ids` file in the *install-dir\mgr* directory.
2. Shared memory.

6.2.1 Daemon Resource Locks

Caché 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 `clock` in the *install-dir/bin* 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 clock 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.

6.2.2 Cache.ids File

The `cache.ids` file contains the name of the node where Caché was started. The existence of the `cache.ids` file acts as a flag to Caché 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 `cache.ids` is being read, it will prevent Caché from starting up.

In previous versions of Caché, the shared memory identifiers were also stored in the `cache.ids` file, but this is no longer the case.

6.2.3 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 `cache.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 Caché).

6.2.3.1 Step 1. Check the Status of the Instance

The startup routine runs `cache -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 `cache.ids` file. If the `cache.ids` file does not exist, no information is available on where the daemons are running.

Action: The user must run `ccontrol stop` or `ccontrol force` to halt the running instance on the appropriate node. This stops the daemons and deletes the `cache.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 `cache.shid` file, which tracks the shared memory ID of Caché on all UNIX® platforms except macOS. Contact the [InterSystems Worldwide Response Center](#) (WRC) for support correcting the file.

6.2.3.2 Step 2. Start Caché

The Caché startup process (`cache`) 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, Caché reports this and exits with an error. Startup halts.

The node on which the daemons are running is unknown if the `cache.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 `cache.ids` file.

Caché continues startup.

6.3 Managing Caché

From the shell, a user with any user ID in the `sysmgr` group can run `ccontrol` (see [Controlling Caché Instances](#) in the “Using Multiple Instances of Caché” chapter) which invokes Caché executables and scripts in the `install-dir/bin` directory. The following sections describe how to perform these management tasks on a Caché instance:

- [Starting Caché](#)

- [Running Caché](#)
- [Stopping Caché](#)

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 non-root 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.

6.3.1 Starting Caché

To start Caché, 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 Caché, view the `cconsole.log` file as described in the [Monitoring Log Files](#) section of the *Caché Monitoring Guide*.

If you are not on the console machine, run Telnet and connect to the target machine where Caché is installed. Before you can start Caché 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 Caché installation as having privileges to stop and start the system.

See the “[Installing Caché on UNIX® and Linux](#)” chapter of the *Caché Installation Guide* for information on specifying such privileges during installation.

Start Caché using the `ccontrol` command:

```
ccontrol start <instname>
```

where *instname* is the name of the Caché instance you want to start. See [Controlling Caché Instances](#) in the “Using Multiple Instances of Caché” chapter of this guide for more options and information.

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

6.3.2 Running Caché

From the shell, a user with any user ID and any group ID (`anyuser:anygroup` in this example), can run `csession` (see [Connecting to a Caché Instance](#) in the chapter “Using Multiple Instances of Caché”) which executes `cuxsession` in the `install-dir/bin` directory.

Running as `anyuser:cacheusr`, Caché 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 **csession**. Thus, while any user may run Caché, the activities of that user once in Caché are defined and limited by the security roles assigned to that user.

CAUTION: Do *not* enter Caché by invoking its executable (`cache.exe`) directly from the `install-dir/bin` directory.

The Caché executable is not itself a setgid-executable. It is the responsibility of the **csession** wrapper to set the group properly on behalf of the user entering Caché. This is not a problem if you are running Caché from the `/usr/bin` directory as set up by the **ccontrol** default function. The **ccontrol** default sets up an executable file called `cache` in the `/usr/bin/` directory and is a link to call **csession** which sets permissions properly.

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

6.3.3 Stopping Caché

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

To stop Caché on UNIX®, the same requirements exist as for starting Caché. 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 Caché installation as having privileges to stop and start the system.

To stop Caché, from the command line:

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

```
ccontrol stop <instname>
```

where *instname* is the name of the Caché instance you want to stop. (See [Controlling Caché Instances](#) in the “Using Multiple Instances of Caché” chapter of this guide for more **ccontrol** options and information.)

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

2. This procedure invokes the Caché **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 Caché continues running.

7

Connecting to Remote Servers

You can control remote instances from a Telnet session, from the **Remote System Access** submenu on the Caché Launcher on Windows, or from a URI generated from the web server and instance information.

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 Caché Launcher and point to **Remote System Access**.
3. Point to a launcher utility and then click the server name.

You may also connect to a remote instance of Caché from a Telnet session:

1. Click the Caché Launcher and point to **Remote System Access**.
2. Click **Caché Telnet**, connect to the remote server, and log on to the Caché system with your username and password. Alternatively, if the server is on the preferred server list, point to **Terminal** and then click the server name.

You can remotely log into a Caché 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 a Caché server instance. The version of Caché on the client machine in most cases, must be the same or a later version of the Caché 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
```

You can also use Caché Server Pages (CSP) with a remote web server. This can be used to manage a Caché instance using the Management Portal.

When connecting to the **Management Portal** or **Documentation** choices on the **Remote System Access** menu, you can bookmark the generated URIs and return to these pages on your web server as long as the remote instance is running.

Important: If you are using the Internet Explorer web browser you may need to adjust the Local intranet security settings to properly connect to remote servers.

7.1 Define a Remote Server Connection

To use the Caché Launcher utilities or other Caché applications on a remote server, the server must be on the connection list in the Caché 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 Caché on that server.

Important: The Caché superserver must be running on the remote machine and its port must be open on your firewall to use the Caché 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 Caché Launcher menu, point to **Preferred Server**, and click **Add/Edit** to open the Caché 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.

Table 7–1: Add Connection Input Fields

Input Field	Description
Server Name	A descriptive phrase that identifies the server; it is what appears as a selection in the Caché 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. Caché 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	<i>Optional</i> – The IP address of the web server you wish to use to manage this Caché instance. Defaults to IP Address if not specified. See Using Optional Server Fields for more information.
Web Server Port	The port number of the web server. The default port number is 57772.
CSP Server Instance	<i>Optional</i> – The Caché instance name to which you want to connect if you are configuring one web server to connect to multiple Caché instances. See Using Optional Server Fields for more information.
Comment	<i>Optional</i> – A description of the remote server.
Authentication Method	Choose Kerberos ¹ or Password ² .
Connection Security Level	For Kerberos ¹ only. Choose either Kerberos , Kerberos with Packet Integrity , or Kerberos with Encryption .
Windows Caché Telnet Server	For Kerberos ¹ only. Select this check box if you are defining a connection to a Windows server.
Service Principal Name	For Kerberos ¹ 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.

¹ See the [Kerberos Authentication](#) section of the “Authentication” chapter in the *Caché Security Administration Guide* for details on the use of these fields.

² See the [Caché Login](#) section of the “Authentication” chapter in the *Caché Security Administration Guide* for information about password-based authentication in Caché.

7.1.1 Using Optional Server Fields

Web Server IP Address

If you plan to use a web server that exists on a separate machine from the Caché instance you are managing, enter the IP address of the web server machine in this field. Defaults to **IP Address** if not specified.

CSP Server Instance

If you are configuring one web server to connect to multiple Caché instances, enter the Caché 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 Caché, cache1 and cache2, by default you manage each instance with its own private Apache web server that is installed as part of Caché.

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 cache1 and cache2 in the **CSP Server Instance** field when creating a server definition for each instance. This automatically creates virtual directories /cache1 and /cache2 on the public web server that point to the corresponding Caché 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/cache1/csp/sys/UtilHome.csp
http://localhost:80/cache2/csp/sys/UtilHome.csp
```

If you plan to use Caché development tools and CSP applications remotely or control multiple instances from one web server, perform the advanced configuration steps described in the following section: [Advanced Web Server Configuration](#).

7.2 Advanced Web Server Configuration

For some Caché 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 Caché 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 Caché 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 Caché 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 [CSP Gateway Configuration Guide](#).

8

Using Multiple Instances of Caché

You can install and run multiple instances of Caché on a single host system. Each instance is a unique, independent Caché environment. This chapter addresses the following topics:

- [Caché Instances](#)
- [Connecting to a Caché Instance](#)
- [Controlling Caché Instances](#)
- [Configuring Multiple Caché Instances](#)

8.1 Caché Instances

There are many ways to connect to a Caché instance. Two of the most common are through the Caché Launcher (only on Windows platforms) and from the command line of the console.

Each Caché instance installed on a Windows machine has its own launcher. All of the launcher utility functions operate within the context of the instance associated with that launcher.

As you install each instance of Caché, you give it an instance name. To perform command-line procedures on a particular instance of Caché, use this name. You can start, stop, and maintain each of these instances independently. The following table shows the most common commands to manipulate Caché instances.

Table 8–1: Control Commands

Action desired	Caché command
Starting a Caché instance	<code>ccontrol start <i>instname</i></code>
Connecting to a Caché instance	<code>csession <i>instname</i></code>
Stopping a Caché instance	<code>ccontrol stop <i>instname</i></code>

These commands are described in detail in [Connecting to a Caché Instance \(csession\)](#) and [Controlling Caché Instances \(ccontrol\)](#).

From the Caché Launcher, you can also control multiple remote Caché environments. Caché allows you to create and edit all Caché data on remote systems. This includes, but is not limited to, running remote backups, editing remote instances,

and creating and compiling remote objects and routines. See the chapter “[Connecting to Remote Servers](#)” in this guide for more detailed information.

Note: On Windows platforms, you can also start, stop, and restart a Caché instance by using the Windows Services administrative tool to start, stop, and restart the Caché Controller service for the instance. However, you cannot use this method to start an instance configured for startup with interactive encryption key activation (see [Configuring Startup with Interactive Key Activation](#) in the “Managed Key Encryption” chapter of the *Caché Security Administration Guide*) because it does not allow you to enter the required credentials.

8.2 Connecting to a Caché Instance

Once you have started Caché, you can log into Caché on the command line using the **csession** command, as follows:

```
csession <instname> [arguments]
```

where *instname* is the name of the instance that you are managing, and *[arguments]* indicates one or more of the optional arguments described in the following table. On a Windows system, you must execute the command from its location, the *install-dir\bin* directory of a Caché instance, or include the full path in the command, for example:

```
C:\InterSystems\Cache27\bin\cssession Cache33
```

Note: As shown in the preceding example, the instance name you specify can be different from the instance of the **csession** binary you are executing, but is always required, even if they are the same.

Table 8–2: Caché csession Command and Arguments

Command	Description
<code>csession instname -B</code>	Provides system administrator emergency login. Note: To execute an emergency startup on Windows, you must open the Command Prompt using the Run as Administrator option. See also Emergency Access in the “System Management and Security” chapter of the <i>Caché Security Administration Guide</i> for information about starting and logging into Caché in emergency access mode.
<code>csession instname -U namespace</code>	Specifies the login namespace. Note: -U has no effect if you are starting Caché with a user account whose Startup Namespace is specified (see Properties of Users in the “Users” chapter of the <i>Caché Security Administration Guide</i>).
<code>csession instname -b partition_size</code>	Specifies the maximum partition size (in KB) for the process.
<code>csession instname "[label[+offset]]^routine"</code>	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: <ul style="list-style-type: none"> • <code>"routine([parameter-list])"</code> • <code>"[label]^routine([parameter-list])"</code> • <code>"##CLASS(package.class).method([parameter-list])"</code> where, for example, <code>parameter-list</code> is specified in the form <code>"string literal",,--000123.45600E+07</code> , and omitted parameters are passed to the target as <code>\$Data(parameter)=0</code> . Note: Whitespace and shell meta characters must be quoted in an operating-system dependent form.

To use the **csession** command or the **ccontrol** command (see [Controlling Caché Instances](#)) on a remote server, use `csession` or Terminal, or a remote connection through Terminal or Telnet/ssh client.

8.3 Controlling Caché Instances

You can control an instance of Caché using the **ccontrol** command. The **ccontrol** command supports a number of functions and has the following syntax:

```
ccontrol <function> <instname> [arguments]
```

where *instname* is the instance name that you chose during the installation.

Note: To display the most current help file for running the **ccontrol** command, invoke **ccontrol help**. On Windows, run the command from the *install-dir*\Bin directory (or include the full path with the command):

```
C:\InterSystems\Cache27\bin>ccontrol help
```

The **ccontrol help** command displays the CcontrolHelp.html file in your browser; the file is in the *install-dir*\Help directory.

The optional arguments (indicated by [*arguments*]) are:

- *quietly* — non-interactive with minimal dialog
- *nostu* — do not run the startup routine (^STU)
- *help* — print the help message and exit
- *restart* — start Caché after a successful shutdown (for **stop** function only and not available on Microsoft Windows platforms)

To use the **ccontrol** command or the **csession** command (see [Connecting to a Caché Instance](#)) on a remote server, use a Telnet or ssh client or the Terminal.

The following table displays some common uses of the **ccontrol** command.

Important: The **ccontrol** command has other platform-specific functions, which can be displayed with the **help** function; functions that are internal to InterSystems do not appear in the help display.

Table 8–3: Caché ccontrol Command and Functions

Command	Description
ccontrol start <i>instname</i> [args]	Starts instance <i>instname</i> . Note: You may be prompted to start in “Emergency Mode;” if so, see Emergency Access in the “System Management and Security” chapter of the <i>Caché Security Administration Guide</i> for more information.
ccontrol start <i>instname</i> [full pathname of .cpf file]	Starts Caché instance <i>instname</i> , optionally using the specified <i>name.cpf</i> file. If a <i>name.cpf</i> file is not specified, the Caché instance uses the <i>cache.cpf</i> file.
ccontrol start <i>instname</i> nostu	On all platforms except Windows, starts the specified instance without running ^STU.
ccontrol startnostu <i>instname</i>	On Windows, starts the specified instance without running ^STU.
ccontrol stop <i>instname</i> [args]	Shuts down an instance.
ccontrol stopnoshut <i>instname</i> ⁴	Shuts down the named instance using INTNOSHUT^SHUTDOWN .
ccontrol stopstart <i>instname</i>	Shuts down and restarts the named instance.
ccontrol force <i>instname</i>	Forces an instance down.

Command	Description
ccontrol list	<p>Lists the following information for all installed instances:</p> <ul style="list-style-type: none"> • Instance name (and installation type) • Installation directory • Caché version • Pathname of Caché parameter (.cpf) file • Superserver and webserver port numbers • Instance status, as follows <ul style="list-style-type: none"> – running – down – starting or stopping – incomplete start or stop, logins disabled • Instance's system health state, if running (see System Monitor Health State in the "Using the Caché System Monitor" chapter of the <i>Caché Monitoring Guide</i>) (not included on Windows) • Mirror member type and status (if a mirror member) (see <code>%SYSTEM.Mirror.GetMemberType()</code> and <code>%SYSTEM.Mirror.GetMemberStatus()</code>)
ccontrol qlist	<p>Lists essentially the same information for each instance as ccontrol list, plus the JDBC Gateway port, but without labels and all on one line, separated by carets (^), in the following order. You can specify a single instance on the command line to display information for that instance only.</p> <ul style="list-style-type: none"> • Instance name (and installation type) • Installation directory • Caché version • Instance status • Pathname of Caché parameter (.cpf) file • Superserver, webserver, and JDBC Gateway port numbers • Instance's system health state, if running (not included on Windows) • Mirror member type and status (if a mirror member)

Command	Description
ccontrol all ¹	<p>Lists the following information for all installed instances, one instance per line:</p> <ul style="list-style-type: none"> • Instance status, as follows <ul style="list-style-type: none"> – <blank> (status unavailable, logins disabled) – dn (down or has crashed) – up (running) – st (starting or stopping) • Instance name • Caché version • Superserver port number • Installation directory
ccontrol allw	Lists the same information for each instance as ccontrol all , without wrapping long field values. Lines longer than 80 characters may result.
ccontrol qall	On platforms other than Windows, lists the same information for each instance as ccontrol all , except that long lines are truncated to 78 characters plus a terminating tilde (~);
ccontrol stat <i>instname</i> [args]	Retrieves system statistics. This is the same as the cstat utility (see the appendix “ Monitoring Caché Using the cstat Utility ” in the <i>C Monitoring Guide</i>). ccontrol stat is not available on Windows, but cstat is.
ccontrol help	Displays most recent information about the ccontrol command.
ccontrol help [start stop force]	Gives function-specific help for the start, stop, and force functions.
ccontrol rename <i>instname</i> { <i>newname</i> }	Renames the instance (not available on Windows systems).
ccontrol [run console cterminal] <i>instname</i>	Runs Caché in programmer mode with either no device, the console, or the terminal for \$Principal (Windows only). ⁵
ccontrol [run console cterminal] <i>instname</i> <i>routine</i>	Runs the named Caché routine in application mode with either no device, the console, or the terminal for \$Principal . (Windows only). ⁵
ccontrol [run console cterminal] <i>instname</i> <i>routine</i> <i>namespace</i> ³	Runs the named Caché routine in the indicated namespace in application mode with either no device, the console, or the terminal for \$Principal . (Windows only). ⁵
ccontrol runw <i>instname</i> <i>routine</i> [<i>namespace</i>] ³	Runs the named Caché routine in the specified namespace (if any) in application mode with no input/output device for \$Principal (Windows only). When run from a batch script, ccontrol waits for the Caché process to terminate before returning. The exit code from the Caché process is returned by ccontrol . ⁵

Table Notes

¹ If you require complete information, such as for parsing or reporting purposes, use **ccontrol list**.

² (removed)

³ The *namespace* argument has no effect if you are starting Caché with a user account whose *namespace* property contains a value. See the “[Users](#)” chapter of the *Caché Security Administration Guide* for details on maintaining user accounts.

⁴ Only owner and cacheusr can run SHUTDOWN without csession login.

⁵ On UNIX®/Linux systems, these tasks are executed using the **csession** command (see [Connecting to a Caché Instance](#)).

8.4 Configuring Multiple Caché Instances

You can install and simultaneously run multiple instances of Caché 4.0 and later on a single machine. Install Caché 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:

- Each instance must be version 4.0 or later of Caché. For example, Caché 5.1 and 3.2.1 cannot run on the same machine at the same time.
- 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 Caché 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 the next section for information on how to [Set Port Numbers](#).

Note: Please see the [Multiple Caché Installation Issues](#) section of the *Caché Installation Guide* for important facts when installing multiple instances of Caché on the Windows platform.

8.4.1 Set Port Numbers

For a standard, single instance of Caché, the superserver port number is 1972 by default. For multiple instances of Caché 4.0 and later on a single machine, each must have a unique port number. During installation, subsequent instances are assigned the next available 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 Caché instance as follows:

- Superserver port number — 1972 or the first available subsequent number equal to or higher than 56773
- [Web server port number](#) — 57772 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 Caché 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 Caché on one machine, verify that each is assigned unique port numbers. The assignment of unique port numbers avoids confusion when using the Caché 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 non-InterSystems tools, 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. In general, InterSystems recommends that server side installations of our products be protected by physical security and isolation. This protection should lessen the need for other tools or at least the frequency with which they are run.

Virus scanner observations:

1. To deliver virus-checked software, InterSystems products are delivered out of a sanitized environment to our customers, and by providing a checksum for verification.
2. Performance impact.

Remedy: For scanning purposes, exclude the following files and directories:

- The WIJ file and the directory* containing the WIJ file (see the chapter “[Write Image Journaling and Recovery](#)” in the *Caché Data Integrity Guide*).
- All database files (.DAT) and directories* containing database files (see [Configuring Databases](#)) in the “Configuring Caché” chapter of this guide.
- Any directory* in which journal files are stored or processed (see the chapter “[Journaling](#)” in the *Data Integrity Guide*).
- Any other file/directory* that is actively required for Ensemble to function. Examples would be the alternate journal directory* (see “[Journaling](#)” in the *Data Integrity Guide*), or any directory* being used by an Ensemble business service or production.

* varies based on configuration settings

3. False positives on InterSystems executables.

Remedy: For scanning purposes, exclude all cache.exe files and directories containing cache.exe files.

WARNING! EXCLUDING ITEMS FROM MALWARE SCANS MAY INTRODUCE VULNERABILITIES INTO PROTECTED DEVICES AND APPLICATIONS. THE CLIENT ASSUMES ALL RESPONSIBILITY FOR CONFIGURING MALWARE PROTECTION.

Finally, when InterSystems observes interactions between third-party software and our products that negatively affect our product behavior, we report those issues to the third-party vendor.

B

Feature Tracker Collects Usage Statistics

As part of InterSystems' ongoing efforts to improve its products in line with customer needs, InterSystems includes a software utility in Caché called Feature Tracker that gathers statistics on software module usage. This appendix describes Feature Tracker.

B.1 Why We Gather Statistics

The statistics track whether or not software modules are present and used in a given Caché instance. Feature Tracker sends this information via https to InterSystems weekly. These statistics help us plan development and support. The information gathered does not include any application data.

B.2 Where Logs Are

If Caché is successful in sending data, it updates the file `FeatureTracker.log` in the manager directory `<installdir>\mgr`. This file contains a JSON-format copy of the data that was sent. Each entry has the following form:

```
"FT.<feature>":<value>,
```

If Cache is unable to send the data, it tries several more times. If it is still unsuccessful, Cache stops trying and tries again at the next regularly schedule time.

Cache updates the console log file (`cconsole.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

The following software statistics are gathered:

Feature Tracker logs whether the following software features are enabled:

- BitTrakCare
- BI User (Runtime) and BI Development
- C-Type License
- DeepSeeRun, DeepSeeAnalyze, DeepSeeModel, DeepSee Visual Reporting Data Definition, DeepSee Visual Reporting Execute, DeepSee Visual Reporting Format Definition
- Ensemble
- ExtremeNoUserLimit
- Healthshare, Healthshare Foundation
- iKnow
- IPNeutral
- Shadow Addon
- Web License

Feature Tracker also logs the following:

- Caché Basic: Compilation and execution
- Database (DB) Encryption: Enabled and number of mounted encrypted databases available at snapshot time
- Mirroring: Reports whether this instance is a member of a mirror, connected to a mirror, number of failover members, number and type of async members, number of mirror sets each async member is a member of
- MultiValue (MV): Command execution, compilation, execution of code
- System: Authorization, Caché version, operating system, host name, instance name, instance ID, 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 still collects 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 Caché, the upgrade preserves the state of Feature Tracker. If it the task was scheduled before the upgrade, it remains scheduled, and, if the task was suspended, it remains suspended.

