



Caché System Administration Guide

Version 2009.1
30 June 2009

Caché System Administration Guide
Caché Version 2009.1 30 June 2009
Copyright © 2009 InterSystems Corporation
All rights reserved.

This book was assembled and formatted in Adobe Page Description Format (PDF) using tools and information from the following sources: Sun Microsystems, RenderX, Inc., Adobe Systems, and the World Wide Web Consortium at www.w3c.org. The primary document development tools were special-purpose XML-processing applications built by InterSystems using Caché and Java.



Caché WEBLINK, Distributed Cache Protocol, M/SQL, M/NET, and M/PACT are registered trademarks of InterSystems Corporation.



InterSystems Jalapeño Technology, Enterprise Cache Protocol, ECP, and InterSystems Zen are 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 Customer Support

Tel: +1 617 621-0700
Fax: +1 617 374-9391
Email: support@InterSystems.com

Table of Contents

About This Book	1
1 Using the System Management Portal	3
1.1 Starting the Portal	3
1.1.1 Portal Login Page	4
1.1.2 Portal CSP Application Structure	5
1.2 Portal Pages Overview	6
1.2.1 System Overview Information	7
1.2.2 Common Portal Procedures	8
1.3 Portal Functions Overview	8
1.3.1 System Administration	8
1.3.2 Data Management	9
1.3.3 Operations	9
2 Configuring Caché	11
2.1 Configuring Data	13
2.2 Configuring Namespaces	14
2.2.1 Create a Namespace	15
2.2.2 Modify Default Database Mapping	15
2.2.3 Add Global, Routine, and Package Mapping to a Namespace	15
2.3 Configuring Databases	20
2.3.1 Local Databases	21
2.3.2 Remote Databases	23
2.4 Configuring System Information	25
2.4.1 Memory and Startup Settings	25
2.5 Configuring Task Manager Email Settings	27
2.5.1 Configure Email Settings	27
2.6 Configuring NLS Settings	28
2.6.1 Using the NLS Settings Page of the System Management Portal	29
2.6.2 Using the ^NLS Routine	31
2.6.3 Using the NLS Class Packages	32
2.7 Cluster Settings	33
2.8 Zen Report Settings	34
3 Managing Caché	35
3.1 Maintaining Local Databases	35
3.1.1 Manage Freespace	36
3.1.2 Check Database Integrity	39
3.1.3 View Integrity Log	40

3.2 Controlling Caché Processes	41
3.2.1 Display Process Information	42
3.2.2 Display Process Details	45
3.2.3 Broadcast Messages to Terminals	47
3.3 Using the Task Manager	48
3.3.1 Schedule New Task	48
3.3.2 Run Task On Demand	50
3.3.3 View Upcoming Tasks	50
3.3.4 View Task Schedule	51
3.3.5 Task Details	51
3.3.6 Task History	52
3.3.7 View Task History	52
4 Managing Caché Licensing	53
4.1 Configuring Caché Licensing	53
4.1.1 Configure License Servers	54
4.2 Updating the License Key	55
4.2.1 License Troubleshooting	56
4.3 Determining License Capacity and Usage	57
4.3.1 Methods to Show Local License Information	58
4.3.2 Methods to Show License Server Information	59
4.4 Identifying Users	60
4.4.1 License Logins	61
4.4.2 Username Licensing	61
4.4.3 CSP Connections and Other Special Cases	62
5 Using Caché on Windows	65
5.1 Caché Cube	65
5.2 Starting Caché	67
5.3 Stopping Caché	68
5.4 Removing the Cube from the System Tray	69
5.4.1 Temporarily	69
5.4.2 Permanently	69
5.5 Controlling Caché from the Command Prompt	69
6 Using Caché on UNIX, Linux, and Mac OS X	71
6.1 UNIX User and Group Identifications	71
6.2 Startup on UNIX	72
6.2.1 Shared Memory Object	72
6.2.2 Daemon Resource Locks	72
6.2.3 Cache.ids File	73
6.2.4 Startup Sequence	73

6.3 Managing Caché	75
6.3.1 Starting Caché	75
6.3.2 Running Caché	77
6.3.3 Stopping Caché	77
6.4 UNIX File Permissions	78
7 Using Caché on OpenVMS	81
7.1 Starting Caché	81
7.2 Stopping Caché	82
7.3 Accessing the System Management Portal	82
7.3.1 Configuring the System Management Portal on Windows	83
7.3.2 Advanced Configuration	84
8 Connecting to Remote Servers	85
8.1 Define a Remote Server Connection	86
8.1.1 Using Optional Server Fields	87
8.2 Advanced Web Server Configuration	88
9 Using Multiple Instances of Caché	91
9.1 Caché Instances	91
9.2 Connecting to a Caché Instance	92
9.3 Controlling Caché Instances	93
9.4 Configuring Multiple Caché Instances	95
9.4.1 Set Port Numbers	96
10 Achieving High Availability	99
10.1 Surviving a Crash	99
10.1.1 Database Integrity Protection	100
10.2 Write Image Journaling	100
10.3 Backup Strategy	101
10.4 Logical Data Protection	101
10.4.1 Transaction Processing	102
10.4.2 Transactions and Locking	103
10.4.3 Global Journaling	103
10.5 Shadow Journaling	104
10.6 System Failover Strategies	105

List of Figures

System Management Portal Home Page	4
Sample Namespace Mapping	16
Caché Cube Menu	66

List of Tables

System Administration Tasks	8
Database Management Tasks	9
System Operation Tasks	9
System Configuration Tasks	11
Connectivity Tasks	12
Additional Tasks	12
Local Databases List Information	35
Local Databases Freespace Information	37
Process Management Functions	42
Process Column Information	43
Possible Process States	43
General Information Indicators	45
Client Application Details Indicators	46
Execution Details Indicators	46
Add Connection Input Fields	86
Control Commands	92
Caché csession Command and Arguments	92
Caché ccontrol Command and Functions	93
Failover Strategy Impact	105

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 System Management Portal 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 Caché Terminal or used to manage a remote system using Telnet.

This book introduces the following topics:

- [Using the System Management Portal](#)
- [Configuring Caché](#)
- [Managing Caché](#)
- [Managing Caché Licensing](#)
- [Using Caché on Windows](#)
- [Using Caché on UNIX, Linux, and Mac OS X](#)
- [Using Caché on OpenVMS](#)
- [Connecting to Remote Servers](#)
- [Using Multiple Instances of Caché](#)
- [Achieving High Availability](#)

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

For general information, see *Using InterSystems Documentation*.

1

Using the System Management Portal

Caché lets you perform system administration and management tasks via a CSP application, the InterSystems System Management Portal (SMP). 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 Portal](#)
- [Portal Pages Overview](#)
- [Portal Functions Overview](#)
- [Portal CSP Application Structure](#)

1.1 Starting the Portal

You can start the System Management Portal in the following ways:

- From the Caché online documentation, click **Management Portal** under the InterSystems logo in the title bar. There are also several active links to relevant portal pages throughout the documentation. For example, if you are reading online documentation now, click **[Home] > [Configuration] > [Memory and Startup]** to navigate to the portal page where you can configure the memory and startup settings for this instance of Caché.

- From your Web browser go directly to the System 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).

- On Microsoft Windows platforms, click **System Management Portal** on the Caché Cube menu.

Depending on the authentication settings for your system you may first have to log in before going to the **[Home]** page. See the [Portal Login Page](#) section for an overview of login requirements.

System Management Portal Home Page



Each functional area requires access to particular resources. See the [Portal CSP Application Structure](#) section for more details.

Important: *Microsoft Windows 2003 Users Trusted Site Security Setting* — The first time you visit the System 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 Portal Login Page

Whether or not you must enter a user name and password to use the portal depends on the authentication settings of the System Management Portal CSP application (`/csp/sys`). There are two conditions:

Only Unauthenticated Access

Neither requires nor accepts a user name and password — If the portal only accepts unauthenticated connections (the default setting for Minimal security installs), you do not require a user name 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 user name and password — If your security settings require authentication for the System Management Portal CSP 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 System Management Portal **[Home]** page.

Important: You can look up the **Authentication allowed** settings on the **[Home] > [Security Management] > [CSP Applications]** page by clicking **Edit** in the /csp/sys application row.

1.1.2 Portal CSP Application Structure

The main portal CSP application is /csp/sys. To provide enforcement of privileges within the System Management Portal, the portal is further split into distinct CSP applications. The System 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	CSP 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 Portal Pages Overview

This section describes the information displayed on the System Management Portal pages.

Portal Title Bar

The left side of the title bar displays the following information:

- **Page name** — displays the name of the current portal page
- **Licensee** — displays the value of the Customer Name field in the cache.key file

Beneath the InterSystems logo are the following buttons:

- **Home** — brings you to the System Management Portal **[Home]** page
- **About** — displays [system overview information](#)
- **Help** — opens the online documentation at an appropriate topic
- **Logout** — brings you to the **Login** page of the portal

The upper right corner of the title bar displays the following information:

- **Server** name — the name of the server running Caché
- **Instance** name — the name of the Caché instance running on the server
- **User** name — the name of the user logged into the portal

Navigation Bar

Throughout the System Management Portal, the navigation bar provides quick links to common functions.

- **Navigation Path** — The left-hand side of the navigation bar shows the navigation path that brought you to the page you are viewing. Each represented page in the navigation has an active link.

For example, the navigation path area for the **Memory and Startup** page contains: **[Home] > [Configuration] > [Memory and Startup]**, which provides links to the **[Home]** page and the **[Home] > [Configuration]** page.

- **Go to box** — a list of common tasks available based on the roles the user holds

1.2.1 System Overview Information

When you click **About** on the title bar of the System 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 and port number where the Caché License server 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
- **Preferred language for this session** — Contains a drop-down list of System Management Portal languages available with the specified locales. If you change this setting, it takes effect immediately and remains for this session.

1.2.2 Common Portal Procedures

There are some functions that are similar on many or all portal pages:

- *Menu pages* — On pages that contain menu items, moving your cursor over a particular item displays a description of that item in a shaded box in the lower left corner of the page.
- *Filter box* — Pages with lists contain a filter box which you can use to shorten the list. Enter any literal in the **Filter** box; the list shortens as you enter more characters, displaying any item that contains that particular string. Using the asterisk (*) symbol is not necessary.

1.3 Portal Functions Overview

The portal is divided into three functional areas:

- [System Administration](#) — System administration tasks
- [Data Management](#) — Database management tasks
- [Operations](#) — System operation 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 Administration

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

System Administration Tasks

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

Menu Item	Documentation Source
Ensemble Management Portal	Managing Ensemble Productions Note: Ensemble installations only.

1.3.2 Data Management

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

Database Management Tasks

Menu Item	Documentation Source
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>

1.3.3 Operations

The system operator tasks are divided into the categories displayed on the **Operations** portion of the System 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.

System Operation Tasks

Menu Item	Documentation Source
System Dashboard	“ Monitoring Caché Using the System Management Portal ” chapter of the <i>Caché Monitoring Guide</i> .
Backup	Running Caché Backups section of the “Backup and Restore” chapter of <i>Caché Data Integrity Guide</i> .
Databases	Maintaining Local Databases section of the “Managing Caché” chapter of this guide.

Menu Item	Documentation Source
Processes	Controlling Caché Processes section of the “Managing Caché” chapter of this guide.
Locks	Monitoring Locks section of the “Monitoring Caché Using the System Management Portal” chapter of the <i>Caché Monitoring Guide</i> .
Journals	“ Journaling ” chapter of the <i>Caché Data Integrity Guide</i> .
Shadow Servers	“ Shadow Journaling ” chapter of the <i>Caché Data Integrity Guide</i> .
Task Manager	Using the Task Manager section of the “Managing Caché” chapter of this guide.
System Logs	Monitoring Log Files section of the “Monitoring Caché Using the System Management Portal” chapter of the <i>Caché Monitoring Guide</i> .
System Usage	Monitoring System Performance section of the “Monitoring Caché Using the System 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 Report	“ Using the Caché Diagnostic Report ” chapter of the <i>Caché Monitoring 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 System 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 the following categories: **System Configuration**, **Connectivity** and **Additional Settings**. These, in turn, are divided into subcategories. This chapter describes some of the topics; other topics have separate chapters or documents as references. See the listed document sources in the following tables for detailed information.

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

System Configuration Tasks

Menu Item	Documentation Source
Memory and Startup	Configuring System Information section of this chapter
Namespaces	Configuring Namespaces section of this chapter
Local Databases	Configuring Local Databases section of this chapter
Remote Databases	Configuring Remote Databases section of this chapter
Database Backup Settings	Configuring Caché Backup Settings section of the “Backup and Restore” chapter of the <i>Caché Data Integrity Guide</i>
Journal Settings	Configuring Journal File Settings section of the “Journaling” chapter of the <i>Caché Data Integrity Guide</i>
Cluster Settings	Configuring Cluster Settings section of this chapter

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

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>
Object/SQL Gateway Definitions	<p>“Using the Caché SQL Gateway” chapter of <i>Using Caché SQL</i>. For detailed information on creating logical connection definitions for JDBC, ODBC and .NET, see:</p> <ul style="list-style-type: none"> • Using the Caché SQL Gateway with JDBC in <i>Using Caché with JDBC</i> • Using the Caché SQL Gateway with ODBC in <i>Using Caché with ODBC</i> • Using the Object Gateway for .NET
CSP Gateway Management	“CSP Configuration” chapter of <i>Using Caché Server Pages</i>
Network Settings	Multiple Network Device Configuration section of the “Caché Cluster Management” chapter of the <i>Caché High Availability Guide</i>

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

Additional Tasks

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

Menu Item	Documentation Source
NLS Settings	Configuring NLS Settings section of this chapter
Source Control Settings	“Using Studio Source Control Hooks” appendix of <i>Using Caché Studio</i>
SQL Settings	Configuration Settings section of the <i>Caché SQL Reference</i>
Startup Settings	Startup Settings of <i>Caché Additional Configuration Settings Reference</i>
Task Manager Email Settings	Configuring Task Manager Email Settings section of this chapter
Zen Report Settings	Configuring Zen Report Settings section of 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 System Information](#)
- [Configuring Task Manager Email 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.

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 *Caché 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.

Perform the following procedures for configuring namespaces from the **[Home] > [Configuration] > [Namespaces]** page of the System Management Portal:

- [Create a Namespace](#)
- [Modify Default Database Mapping](#)
- [Add Global, Routine, and Package Mapping to a Namespace](#)

The size of the namespace table is automatic and no longer configurable.

2.2.1 Create 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** on the **[Home] > [Configuration] > [Namespaces]** 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.
2. Choose whether the default database is **local** or **remote**.
3. **Select an existing database** for the default mapping of this namespace, or click **Create New Database**
4. Select the **Create a default CSP application for this namespace** check box if you are creating a CSP application that accesses this namespace.
5. You can also **Copy namespace mappings from** an existing namespace.
6. After entering the required information, click **Save** to add your namespace to the configuration.

2.2.2 Modify Default Database Mapping

You can change the database to which your namespace is mapped without restarting Caché:

1. Navigate to the **[Home] > [Configuration] > [Namespaces]** page to see a list of defined namespaces.
2. Click **Edit** in the row of the namespace you wish to modify.
3. Choose the **Default Database for Globals**, the **Default Database for Routines**, and the **Default Database for Temporary Storage** from the list of defined databases.
4. Click **Save**.

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

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

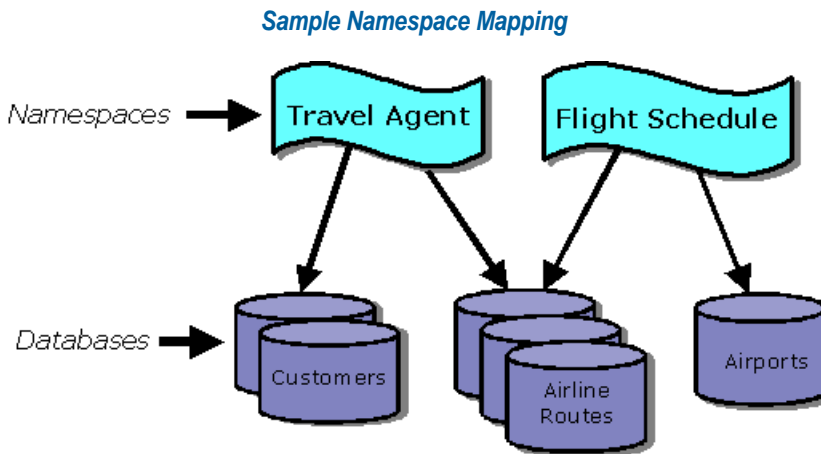
In addition to having access to the globals and routines in the mapped database, you can also map globals, routines, and class packages from other databases on the same or different systems. This allows simple references to data which can exist anywhere and is the primary feature of a namespace. You can map whole globals or pieces of globals; this feature allows data to easily span disks.

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:



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

2.2.3.1 Global Mappings

You can add mapping 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 **[Home] > [Configuration] > [Namespaces]** page, 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 to choose multiple globals.
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" )  
( 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.

6. Click **Advanced** to display the following:

- a. Select the **Collation**.
- b. Select the **Lock Database Location**.

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**.

2.2.3.2 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 **[Home] > [Configuration] > [Namespaces]** page, and click **Routine Mappings** in the row of the namespace where you want to map the global.
2. From the **[Home] > [Configuration] > [Namespaces] > [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. Select the **Routine type**.

6. Click **OK**.

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

7. 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, you would navigate to the **[Home] > [Configuration] > [Namespaces] > [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: If the routines in one namespace (say, N1) are mapped to refer to another database (say D2), then the *^mcq* and *^mqh* globals in the namespace being mapped (N1) also need to be mapped to that other database (D2) as well.

2.2.3.3 Package Mappings

You can add a class package mappings which makes all the classes within a package in a specific database visible to another namespace:

1. Navigate to the **[Home] > [Configuration] > [Namespaces]** page, and click **Package Mappings** in the row of the namespace where you want to map the global.
2. From the **[Home] > [Configuration] > [Namespaces] > [Package Mappings]** page, click **New Package Mapping**.
3. Select the **Package database location** database where the routine 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 **[Home] > [Configuration] > [Namespaces] > [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:

2.2.3.4 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 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.

2.3 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.

Caché databases dynamically expand as needed, though you can set a maximum limit. Secondary volumes (extents) of a database are called cache.ext files. Caché automatically mounts the databases, which can grow to be 32 terabytes if you are using the default 8-KB block size.

The theoretical maximum for the number of Caché databases is 15,998. However, you can only have open at one time as many as permitted by the operating system limit on maximum open files (either per process or system-wide), minus what Caché reserves for its own use and device, which is approximately half.

You can make most database configuration changes dynamically; you can create, delete, or modify database attributes while the system is running. You can create new databases with an 8-KB block size; you can also create 2-KB databases for backward compatibility with versions of Caché prior to release 4.1.

Caché provides wizards for both local and remote database creation. Each is described in the following sections:

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

Considering Endianness

If you are defining a database using an existing cache.dat that was transferred from another platform, you must take Endianness — which defines the order in which multi-byte data is stored — into account. Depending on the platform (see “[Platform Endianness](#)” in *Supported Platforms*), multi-byte data is stored either with 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.” If the cache.dat was created on

a platform whose Endianness is different from the platform you are using, use the [cvendian utility](#) (see the [Using cvendian to Convert Between Big-endian and Little-endian Systems](#) article) to convert the database before you use it.

2.3.1 Local Databases

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

- [Create Local Database](#)
- [Edit Database Properties](#)
- *Delete a Database* — Click **Delete** in the appropriate row to delete the selected database configuration.
- *View Global Data* — Click **Globals** in the appropriate row to view the defined globals in the selected database and their contents.

2.3.1.1 Create Local Database

To create a local database, navigate to the **[Home] > [Configuration] > [Local Databases]** page of the System Management Portal.

1. Click **Create New Database** to open the **Database Wizard**.
2. Enter the following information for the new database:
 - Enter a database name in the text box. Database names must be at least one character (but not more than 30 characters) long. They can start with an alphabetic character or an underscore; the remaining characters must be alphanumeric, dashes, or underscores.
 - Enter a directory name or click **Browse** to select a database directory. If this is the first database you are creating, you must browse to the parent directory in which you want to create the database; if you created other databases, the default database directory is the parent directory of the last database you created.

Important: Caché does not support logical names for database directories on OpenVMS systems.

 - If encryption is activated, you may encrypt this database by selecting the **Encrypt Database** check box.
3. Click **Next** to continue configuring the database or **Finish** to accept the defaults
4. In the **Initial Size** box, type the number of megabytes for your database size (the default is 1 MB). All new databases are created with a **Block Size** of 8 KB.
5. Click **Next**.

6. 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*)
7. Click **Next** to view a list of the database attributes.
8. 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.

2.3.1.2 Edit Local Database Properties

Click **Edit** to view and change the following database properties:

- **Name**
- **Directory**
- **Encrypted**
- **Block Size (Bytes)**
- **Size (MB)**
- **Maximum Size (MB)** — 0 for Unlimited
- **Expansion Size (MB)** — 0 for Default
- **Resource Name**
- **Global Journal State**
- **New Global Collation**
- **New Global Growth Block**
- **New Global Pointer Block**
- **Stream Location** — Click **Browse** to select the directory in which streams associated with this database are stored. By default, the stream location for a local databases is a subdirectory named `stream` in the `install-dir\Mgr\directory` identified at the beginning of this database properties page (for example, `install-dir\Mgr\directory\stream`).

Note: InterSystems recommends that you use the default location.

- **Preserve global attributes on delete**

- **Read Only**
 - **Mount Required at Startup** — Select **Yes** from the drop-down list to mount the database each time you start Caché.
- Note:** By default, some databases (for example, USER and SAMPLES) are not mounted when Caché starts. This property lets you override the default. For additional information, see The Local Databases List Information table in [Maintaining Local Databases](#) in the “Managing Caché ” chapter of this guide.

The “[Create a Local Database](#)” section describes many of these fields.

2.3.2 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:

- [Create Remote Database](#)
- [Add Remote Database](#)
- [Edit Remote Database Properties](#)
- *Delete a Remote Database* — Click **Delete** in the appropriate row to delete the selected database configuration.

2.3.2.1 Create Remote Database

To define a remote database on the local server, perform the following steps:

1. If you have not defined the remote server on this instance of Caché, navigate to the **[Home] > [Configuration] > [ECP Settings]** page of the System Management Portal.
2. Click **Add Remote Data Server** and enter the following information for the ECP remote data server:
 - a. **Server Name** — Enter a logical name for the 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 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.
 - 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*.
5. Navigate to the **[Home] > [Configuration] > [Remote Databases]** page of the System Management Portal.
6. Click **Create New Remote Database** to invoke the **Database Wizard**, which displays a list of the logical names (the name you used when you added it to the list of ECP data servers) of the remote servers on the local server.
7. Click the name of the appropriate server and click **Next**.
8. The portal displays a list of database directories on the remote server. Select one of these to serve as the remote database.
9. Enter a database name (its name on the local server; it does not need to match its name on the remote server) and click **Finish**. 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.

2.3.2.2 Add Remote Database

You can also define a remote database by entering your own remote server and directory. The portal does not validate your input for the following entries:

- **Database Name**
- **Remote Server**
- **Remote Directory**

Once you add a remote database, you can edit its properties or delete it.

2.3.2.3 Edit Remote Database Properties

Click **Edit** to view and change the following database properties:

- **Database Name**
- **Remote Server** — Click the **Available Servers** drop-down to change the remote server where the database resides.
- **Remote Directory** — Click the **Available Directories** drop-down to change the directory on the remote server where the database resides.

- **Stream Location** — Click **Browse** to select 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*) directory.

Note: InterSystems recommends that you use the default location.

The [Create Remote Database](#) section describes many of these fields.

2.4 Configuring System Information

Caché stores system-wide configuration information in a configuration file. By default, the file is stored in the Caché root directory with the .cpf file extension. Caché initially starts with the provided configuration file called Cache.cpf. In most cases, this configuration file is the only one you need to use Caché.

There are a few [Memory and Startup Settings](#) you may want to review on a new installation.

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. See the [Caché Parameter File Reference](#) for more information on the .cpf file.

2.4.1 Memory and Startup Settings

When you first install Caché, you may change some default system information. The **[Home] > [Configuration] > [Memory and Startup]** page of the System Management Portal provides an interface to the database allocation features, as well as a few startup settings:

1. You can choose whether to configure memory automatically or manually.

If you choose **Manually**, you can specify how global buffer pool memory is allocated among 2-KB and 8-KB block sizes.

If you choose **Automatically**, the system allocates the amount of memory best suited to Caché given the available memory.

See the [Memory Configuration](#) section for details.

2. 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.

3. If you select the **Enable Long Strings** check box, Caché allocates a large string stack to handle long strings for each process.
4. If you are running in an IPv6 network, you can select the **IPv6 Enabled** check box to indicate that this instance of Caché accepts IP addresses in IPv6 format as well as IPv4 and DNS format. See the [IPv6 Support](#) section for additional information.

Note: The **IPv6 Enabled** check box is not displayed if the platform on which you are running Caché does not support IPv6 networking.

5. You can change the **Superserver Port Number** for this Caché instance.
6. 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 one field that requires a restart, no change to your configuration takes effect until a restart, even those that normally do not require a restart.

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

`WARNING: There are configuration changes saved that require system restart to take effect.`

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

2.4.1.1 Memory Support

Caché allocates half the space specified by 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.

Important: If you are configuring a large ECP system, allocate at least 50 MB of 8-KB buffers for ECP control structures in addition to the 8-KB buffers required to serve your 8-KB blocks over ECP. See the [Memory Use on Large ECP Systems](#) section of the “Developing Distributed Applications” chapter of the *Caché Distributed Data Management Guide* for details.

2.4.1.2 IPv6 Support

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.

Caché clients accept IPv4, IPv6, and 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, an IPv4 connection is attempted first; then, if an IPv4 connection cannot be made, an IPv6 connection is attempted.

For more information about IPv6 addressing, see [IP Version 6 Addressing Architecture \(RFC 4291\)](#).

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 of the “[Managing Caché Licensing](#)” chapter.

2.5 Configuring Task Manager Email Settings

From the **[Home] > [Configuration] > [Task Manager E-mail Settings]** page of the System Management Portal you can [Configure Email Settings](#) the Task Manager uses for notification.

For more information, see “[Task Manager Email Settings](#)” in the of *Caché Additional Configuration Settings Reference*.

2.5.1 Configure Email Settings

From the **[Home] > [Configuration] > [Task Manager E-mail Settings]** page you can configure email settings the Task Manager uses for notification. Fill in the necessary information from your particular mail server for the following fields:

- **SMTP Server** — address of your outgoing SMTP (Simple Mail Transfer Protocol) mail server
- **SMTP Auth User** and **Password** — Only required for SMTP authentication with the SMTP server. See [RFC 2554](#) for details.
- **Sender** — email address to appear in the sender field
- **Reply To** — email address to which the recipient should reply

The following boxes contain suggested formats; you can change them as you wish:

- **Success Subject** — The formatted subject line of a successful task message
- **Success Message** — The formatted message sent after a successful task runs
- **Failure Subject** — The formatted subject line of a failed task message
- **Failure Message** — The formatted message sent after a task fails

Format the information in the subject and message text boxes using the task parameters listed at the bottom of the page and described in the following table.

Task Parameter	Description
ID	Task ID
DESCRIPTION	Task description
NAME	Task name
LASTSTARTED	Last time the task started
LASTFINISHED	Last time the task finished
SCHEDULED	Last scheduled starting time
CURRENTDATE	Date the email sent
CURRENTTIME	Time the email sent
STATUS	Return value of the task
TASKCLASS	Task class used for this task; for example, %SYS.Task.IntegrityCheck for a database integrity check task,
ERROR	Error code if task failed
SUCCESS	Completed message if task ran successfully

2.6 Configuring NLS Settings

This release of Caché contains updated mechanisms for maintaining National Language Support (NLS) settings. The following sections describe the three ways to manage these settings:

- [Using the NLS Settings Page of the System Management Portal](#)
- [Using the ^NLS Routine](#)

- [Using the NLS Class Packages](#)

CAUTION: InterSystems provides both the **[Home] > [Configuration] > [NLS Settings]** page of the System Management Portal and the **^NLS** routine for you to browse existing locales and tables and for you to create custom locales. Do not edit the system locales and tables (those in the InterSystems product distribution kit), as the next product update will overwrite any changes you make.

For more information, see “[National Language Support Settings](#)” in the of *Caché Additional Configuration Settings Reference*.

2.6.1 Using the NLS Settings Page of the System Management Portal

You can configure many NLS settings using the System Management Portal. The options are parallel to those in the **^NLS** routine.

From the **[Home] > [Configuration]** page, click **NLS Settings** in the **System Configuration** column to display the **[Home] > [Configuration] > [NLS Settings]** page.

You have two menu options under the **Locale Definitions** column:

- [Locale Definitions](#)
- [Import Locale](#)

2.6.1.1 Locale Definitions

From the **[Home] > [Configuration] > [NLS Settings] > [Locale Definitions]** page, you can select a locale and perform several actions (the default of the locale field is always the current locale). Click the appropriate option in the following list:

- **Validate** — Displays a message indicating the validation is successful or an appropriate error message if it is not.
- **Copy** — Enter a locale name in which to create the copy. The new locale name 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. Click **OK** to add the copy to the drop-down list of locale names or **Cancel** to close the dialog box and return to the **Locale Definitions** page.
- **Export** — Enter the file name to receive the export; it must be an **.xml** file. The default name is **loc_%locale.xml**, where **%locale** is the selected locale. In addition, you can include the *pathname* where you want to save the export file; if you do not specify the *pathname*, the default location is *install-dir*\Mgr directory. Click **OK** to export the **.xml** file or **Cancel** to close the dialog box and return to the **Locale Definitions** page.

- **Install** — Select a locale to install that is different from the current locale. An initial validation occurs. If it fails, an error message displays; if it succeeds click **Yes - Install Now** or **Cancel** to close the dialog box and return to the **Locale Definitions** page. You can still install the locale if only a warning is displayed.
- **Load Table** — 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** — Disabled if the selected locale is the current locale. If you select another locale, a confirmation displays. Click **OK** to confirm that you want to delete the locale or **Cancel** to cancel the request and return to the **Locale Definitions** page.

To view and edit details of a selected locale, click **Properties**. The next page displays the locale properties grouped into categories. For each category you can edit the fields and click **Save**, or click **Return** at the top of the page to cancel any of your edits and return to the **Locale Properties** page. The properties are grouped into the following tables:

- **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 the **>** 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**

2.6.1.2 Import Locale

From the **[Home] > [Configuration] > [NLS Settings] > [Import Locale]** page, you can import locales or tables:

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 Routine

Invoke the ^NLS routine from the %SYS namespace:

```
Do ^NLS
```

Follow the series of menus to perform the various functions in managing the NLS locales and tables. The following is an example of using the routine to display the current locale:

```
%SYS>Do ^NLS
```

```
1) Locale definitions
2) Table definitions
3) Import Locales/Tables
4) Current system settings
```

```
NLS option? 1
```

```
Select a locale: enuw => English, United States, Unicode
```

```
1) Display locale
2) Edit locale
3) Install locale
4) Export locale
5) Validate locale
6) Copy locale
7) Delete locale
8) Load locale table
9) Select another locale
```

```
Locale option? 1
```

```
Number of lines for paging (0=don't page): 24 => 0
```

```
----- Locale enuw -----
      Country: United States (US)      Language: English (en-US)
      Character set: Unicode           Currency: $

Internal tables -- Default ----- Additional -----
Pattern match    Latin1
Identifier       Latin1
Uppercase        Latin1
Lowercase        Latin1
Titlecase        Latin1
Collation        Cache standard
$X/$Y action     Latin1

Input/output tables -----
```

Configuring Caché

CP1250	CP1251	CP1252	CP1253	CP1255	CP866
CP874	Latin2	Latin9	LatinC	LatinG	LatinH
LatinT					

I/O defaults -----

Process:	RAW	TCP/IP:	RAW
Cache Terminal:	UTF8	DSM-DDP:	RAW
Other terminal:	RAW	DTM-DCP:	RAW
File:	RAW	System call:	RAW
Magtape:	RAW	Printer:	RAW

----- Date ----- Time ----- Numbers -----

Format: 1	Format: 1	Minus sign: -	Decimal separator: .
Separator: /	Separator: :	Plus sign: +	Group separator: ,
	Precision: 0		Group size: 3

Strings -----

Time -----

AM: AM	Noon: NOON
PM: PM	Midnight: MIDNIGHT

Week days -----

Names:	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Abbrv:	Sun	Mon	Tue	Wed	Thu	Fri	Sat

Months -----

Names:	January	February	March	April	May	June	July	August	September	...		
Abbrv:	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec

- 1) Display locale
- 2) Edit locale
- 3) Install locale
- 4) Export locale
- 5) Validate locale
- 6) Copy locale
- 7) Delete locale
- 8) Load locale table

2.6.3 Using the NLS Class Packages

The [System Classes For National Language Support](#) article 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 *Caché Class Reference*.

2.7 Cluster Settings

This setting applies only on platforms that support clusters (for example, OpenVMS or Tru64). It 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 following settings appear in the Cluster Settings category on the **[Home] > [Configuration] > [Cluster Settings]** page:

- **CommPort** — The port number to advertise in the PIJ to the other cluster members. It is a port number in the range 0–65535.

This setting applies only on platforms that support clusters, for example OpenVMS or Tru64.

- **JoinCluster** — True or false. When true, this configuration is part of a cluster and the appropriate cluster definition settings can be configured. The default is false.
- **Network Type** — Cluster network type. It indicates the cluster network type for this cluster:
 - **Raw Ethernet**
 - **UDP**
 - **ECP** (the default)

Caché clusters require an ECP or DCP network connection between all of the cluster members. The protocol must be the same for all members of the cluster and the required protocol is defined for the cluster when the first member starts and registers the protocol in the pre-image journal.

Note: InterSystems strongly recommends ECP for clusters. When you use ECP, 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** and **CommPort** to force Caché to use a specific IP address and port for the cluster ECP traffic.

The **Raw Ethernet** and **UDP** options imply a DCP connection. The first DCP network port in the configuration is used as the cluster network port. **Raw Ethernet** is only available on OpenVMS.

- **CommIPAddr** — IP Address to advertise in the PIJ to other cluster members to use for inter-cluster communication. The string, which may be up to 128 characters long (a cluster feature), is a resolvable DNS name or IP address.

- **pjdir** — The pre-image journal (PIJ) file to support cluster failover, recovery, and write image journaling in a cluster.

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

For more information, see [Configuring a Caché Cluster](#) in the “Caché Cluster Management” chapter of the *Caché High Availability Guide*.

2.8 Zen Report Settings

From the **[Home] > [Configuration] > [Zen Report Settings]** page you can configure and verify Zen report settings:

- **Path and file name to be used for PDF generation:** — Click **Browse** to locate the command file that invokes the third-party rendering tool that Caché uses to create PDF output. Examples are RenderX XEP and Apache FOP.
- **Verify Now** — Click this button to test whether or not the rendering tool is configured correctly.

For more information, see “[Zen Report Settings](#)” in the of *Caché Additional Configuration Settings Reference*.

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](#)

3.1 Maintaining Local Databases

You can accomplish many database and global management tasks from the **[Home] > [Databases]** page of the System Management Portal. The portal displays a list of local databases. For each database, you see the following information:

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	The status of the directory, which indicates if the database is mounted (including which permissions it has) or unmounted.
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	<p>Action buttons that let you explicitly dismount or mount a database.</p> <p>Note: By default, when you start Caché, some databases (for example, USER and SAMPLES) are not mounted (that is, they are unmounted). When you access an unmounted database that has not been explicitly dismounted, it is mounted dynamically (and it remains mounted until you stop Caché); if, however, you explicitly dismount a database, you cannot make changes to it (that is, a dismounted database is not mounted dynamically when you access it). To override the default, you can configure Caché to mount the database at startup (see Edit Database Properties in the “Configuring Caché” chapter of this guide).</p>

From this page you can perform the following database tasks:

- [Manage Freespace](#)
- [Check Database Integrity](#)
- [View Integrity Log](#)

All the data structures used by Caché are self-balancing and do not suffer performance degradation over time. It is never necessary to take a system down to rebuild it nor to compress data or indices to regain performance.

3.1.1 Manage Freespace

You can perform the following actions relative to the free space on databases through the **[Home] > [Databases]** page of the System Management Portal:

- [Check Database Freespace Information](#)
- [Remove Unused Space](#) (Cleanup)
- [Show Free Space Using %FREECNT](#)

3.1.1.1 Check Database Freespace Information

You can display a page that shows a list of local databases with information about the amount of free space on each local database, and remove unused space on local databases (see [Remove Unused Space](#)). To display the Local Database Freespace Information:

- Navigate to the **[Home] > [Databases]** page of the System Management Portal.
- Click **Freespace** to display a page that shows a list of local databases with the following free space information:

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 size allocated 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 to use system defaults (12% of the current size or 10 MB, whichever is larger).
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 which permissions.

You can retrieve similar information using the **%FREECNT utility**.

3.1.1.2 Remove Unused Space

You can remove unused space on local databases:

1. Navigate to the **[Home] > [Databases]** page of the System Management Portal.
2. Click **Freespace** to display a page (**[Home] > [Databases] > [Freespace]**) that shows a list of local databases and associated freespace information.

- Click **Cleanup** in the row of the database on which you want to remove unused space, and to display a page ([**Home**] > [**Databases**] > [**Freespace**] > [**Cleanup**]) that shows the name, location, and current size of the selected database.

Note: The **Size** displayed in the Cleanup dialog box is the current size of the database expressed in megabytes (MB).

- In the Cleanup dialog box that is displayed, enter the **Target File Size (MB)** and click **OK**. Specify the target file size (in MBs), which must be less than the current size; to remove all unused space, specify 0.
- When the Cleanup 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 Cleanup Database Space background tasks ([**Home**] > [**Background Tasks**]).
- In the Cleanup dialog box, click **Done** to redisplay the Freespace page ([**Home**] > [**Databases**] > [**Freespace**]) that shows a list of local databases (see [Check Database Freespace Information](#)).

Note: If >> is displayed in the first column of the database row, refresh the page to see the new size displayed in the **Size** column.

3.1.1.3 Show Free Space Using %FREECNT

Caché also provides an additional tool to monitor the storage growth of your databases. This utility is called **%FREECNT**. Its output shows the free space available in a database. From a namespace other than %SYS, the utility shows the free space of the databases in that namespace:

```
USER>Do ^%FREECNT
```

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

```

                                Cache Database Free Space
                                Feb 15 2008  7:28 PM
Database      Max Size  Size  Available %Free  Disk Free
c:\MyCache\mgr\user\  Unlimited 1MB   0.52MB   52    42.72GB
```

From the %SYS namespace, you may choose to display the free space of all databases by entering an asterisk (*) at the prompt, or enter one database directory name:

```
%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\cachetemp\
c:\MyCache\mgr\docbook\
c:\MyCache\mgr\samples\
c:\MyCache\mgr\user\
Device:
Right margin: 80 =>
```

	Cache Database	Free Space			
	Feb 15 2008	7:25 PM			
Database	Max Size	Size	Available	%Free	Disk Free
c:\MyCache\mgr\	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 Check Database Integrity

You can check the integrity of databases, or globals stored in a database.

WARNING! Do not run **Integrity Check** or the **^Integrity** utility on a volatile database. Performing an Integrity Check while one or more processes are updating the database may result in the false reporting of database integrity errors.

To check the integrity of databases or globals stored in a database, do the following from the **[Home]** > **[Databases]** page of the System Management Portal:

1. Click **Integrity Check** to display a list of database names with check boxes.
2. Select the appropriate check boxes for the databases you want to check.

Note: If you want to check globals stored in a database, select only the database that contains the globals you want to check, then click **Select Globals** to display a list of globals stored on the selected database. From the list of globals, select the globals you want to check, then click **Save**; if you do not select any globals from the list, all globals in the selected database are checked.

3. Enter the name and location of a log file. You can accept the default (integ.txt), click **Browse** to choose an existing file, or enter your own file name. The integrity check process runs in the background and saves the results to the file name in the text box.
4. Click **OK** to begin the integrity check.

3.1.2.1 Database Integrity Checking Utility

You can also use the utility routine **^Integrity**. The name is case-sensitive and you call it from the Caché manager's namespace with **Do ^Integrity**. This is the equivalent of running **Integrity Check** from the **[Home] > [Databases]** page of the System Management Portal.

This routine includes the following additional entry points:

- **Do CheckPointer^Integrity** asks for a directory and a pointer block at which to start checking.
- **Do Silent^Integrity(logfilename,dirlist)** starts a background process that does an integrity check on selected or all databases and puts the output in a file specified by the *logfilename* parameter. The optional *dirlist* parameter (\$LIST format) identifies selected databases to check; if *dirlist* is not specified, all databases are checked.

This is the equivalent of doing **Integrity Check** from the **[Home] > [Databases]** page of the System Management Portal.

- **Do Query^Integrity(logfilename,outdevice)** does not run an integrity check, but puts the contents of the file specified by the *logfilename* parameter, the results saved from a previous run, out on the current device or the device specified in the optional parameter *outdevice*.

If not specified, *outdevice* is the current terminal. Examples of *outdevice* are a printer, another display device, or another operating system file name. The latter makes a copy of *logfilename*.

- **Do SilentGlobalCheck^Integrity(logfilename,dirlist,gbllist)** starts a background process that does an integrity check on selected globals in a selected databases and puts the output in a file specified by the *logfilename* parameter. The *dirlist* parameter identifies the database that contains the globals you want to check. The *gbllist* parameter (\$LIST format) identifies the globals to check; if *gbllist* is not specified, all globals stored in the specified database are checked.

This is the equivalent of choosing **Select Globals** when doing **Integrity Check** from the **[Home] > [Databases]** page of the System Management Portal.

3.1.3 View Integrity Log

You can view the log resulting from the background process:

1. From the **[Home] > [Databases]** page of the System Management Portal, click **Integrity Log**.
2. Enter the file name used in the Integrity Check process. The default named file displays if you used that name. Otherwise click **View File** to display your integrity log file.
3. You can search for strings within the log file. Enter the string and click **Search**. Matching strings are highlighted.

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 a Caché ObjectScript **Job** command.
- 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 System Management Portal:

Process Management Functions

Function	How to access function from the portal
Display process information	Click Processes under the operations column of the Home page to display the [Home] > [Processes] page.
Display process details	Click Processes under the operations column of the Home page. Click Details in the right hand column of the selected process to display the [Home] > [Process Details] page.
Suspend/resume a process	Click Processes under the operations column of the Home page. Click Details in the right hand column of the selected process to display the [Home] > [Process Details] page. Then click Suspend or Resume (on the operations bar), as desired.
Terminate a process	Click Processes under the operations column of the Home page. Click Details in the right hand column of the selected process to display the [Home] > [Process Details] page. Then click Terminate or Terminate with <RESJOB> Error (on the operations bar), as desired.
Display process variables	Click Processes under the operations column of the Home page. Click Details in the right hand column of the selected process to display the [Home] > [Process Details] page. Then click Variables (on the operations bar) to display the [Home] > [Process Variables] page.
Broadcast messages to terminals	Click Processes under the operations column of the Home page. Click Broadcast (on the options bar) on the [Home] > [Processes] page.

3.2.1 Display Process Information

To display all the active processes on the system and basic information about each, click **Processes** in the **Operations** column of the System Management Portal **[Home]** page. The **[Home] > [Processes]** page displays a table of the processes with statistics about each in columns.

The following table describes the process information available for display:

Process Column Information

Column Heading	Definition
PID	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.
Lines	Number of lines of source code executed.
Globals	Number of global references, including updates, executed (database reads and writes) since the process entered Caché.
State	Process state. See the Possible Process States table below 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.
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 interface](#) is a Caché SQL facility that lets you execute and evaluate Caché ObjectScript commands and expressions from within C programs. You can also use the callin interface from \$ZF routines.

The following table lists the possible process states as displayed in the process information table:

Possible Process States

State	Process
LOCK	Performing a Caché lock.
OPEN	Opening a file or device.

State	Process
CLOSE	Closing a file or device.
USE	In a Use command.
READ	Reading from a file or device.
WRITE	Writing to a file or device.
GGET	Retrieving a value from a global.
GSET	Setting a value into a global.
GKILL	Killing a global/node.
GORDER	Executing \$order .
GQUERY	Executing \$query .
GDEFINE	Executing \$data .
ZF	Calling out from Caché.
HANG	Suspended (by the Caché Hang command).
JOB	Executing a process initiated by the Job command.
JOBEXAM	Examining the state of another running process.
BROADCAST	Broadcasting a message to other processes.
SUSPEND	Suspended (through user interaction using JOBEXAM).
GINCR	Executing \$increment .
GBITSET	Setting bitstrings in a global (Set \$bit).
GBITGET	Retrieving bit values from a global.
RUN	Executing Caché code other than global module or I/O commands.

The following codes may be attached to the end of a state:

- H — Halting
- NH — Network hardening
- N — Remote network
- W — Hibernating

3.2.2 Display Process Details

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

1. Click **Processes** under the **Operations** column of the **[Home]** page.
2. Click **Details** in the row of the appropriate process. This option only exists on processes that you have authority to maintain.
3. The portal displays the **Process Details** page for the process you select.

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

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

General Information Indicators

Indicator	Definition
Process ID	Process ID (PID) number of this process.
User Name	Name of the user currently logged in for this process.
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.
Lines Executed	Number of lines of code executed by this process.
Memory Limit	Amount of memory (Kbytes) allocated for use by this process.
Private Global Blocks	Number of private global data blocks used by this process.
Memory in Use	Amount of memory (Kbytes) currently in use by this process.
Current Device	Name of the I/O device currently in use by this process.
Open Devices	List of devices currently opened by this process.
Lock	Lock information for this process. Click the link at top of the detail box for additional details (mode, counts, and full reference).

Client Application Details Indicators

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).

Execution Details Indicators

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 System Management Portal:

1. Click **Processes** under the **Operations** column of the **[Home]** page.

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 System Management Portal:

1. Click **Processes** under the **Operations** column of the **[Home]** page.
2. Click **Details** in the row of the appropriate process. This option only exists on processes that you have authority to maintain.
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.
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 **[Home] > [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 System Management Portal:

1. Click **Processes** under the **Operations** column of the **[Home]** page.
2. Click **Details** in the row of the appropriate process. This option only exists on processes that you have authority to maintain.
3. The portal displays the **Process Details** page for the process you selected.
4. Click **Variables** on the options bar.

3.2.3 Broadcast Messages to Terminals

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

The utility temporarily takes control of each terminal as it sends the message. Once the terminal receives the message, the previous process continues. The message appears on the terminal screen; it may disrupt

the screen display, but it does not affect user input. The message does not appear in windows running Caché utilities.

To broadcast a message to the terminals associated with a selected process, from the **Operations** column of the **[Home] > [Processes]** page of the System Management Portal:

1. Click the **Processes** menu item.
2. From the page that lists the active processes, click **Broadcast**.
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**.

3.3 Using the Task Manager

From the **[Home] > [Task Manager]** page of the System Management Portal you can choose options to schedule a new task, execute tasks configured as on-demand, view upcoming task manager activities, and view a history of completed tasks.

The following options are available from the **Task Manager Activities** menu:

- [Schedule New Task](#)
- [Run On-demand Task](#)
- [View Upcoming Tasks](#)
- [View Task Schedule](#)
- [View Task History](#)

3.3.1 Schedule New Task

Choosing this 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 the following tasks

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 for the following additional information:

- **Directory**
- **Filename**
- `CumuIncrDBList` — a cumulative backup of databases in the defined list
- `FreeSpace` — a report of free space available on databases in the namespace
- `FullAllDatabases` — a full backup of all databases
- `FullDBList` — a full backup of databases in the defined list
- `IntegrityCheck` — an integrity check of databases in the namespace
- `IncrementalDBList` — an incremental backup of databases in the defined list
- `PurgeJournal` — purge journal files that meet the purging criteria
- `PurgeTaskHistory` — purge task manager history files
- `RunLegacyTask` — run a legacy task; enter the code to execute in the next box
- `ShadowPurge` — purge shadow files
- `SwitchJournal` — switch to a new journal file
- **Task priority** — choose from **Normal**, **Low**, or **High** priority
- **Run task as this user** — choose from the list of defined users
- **Output file** — If the task creates output, you can indicate the file where the task places the output.
- **Reschedule task after system restart?** — **No** or **Yes** indicating if you want to reschedule the task if the system restarts.
- **Send completion email notification to** — If you have configured email settings for the Task Manager, indicate the email address to send successful notification.
- **Send error email notification to** — If you have configured email settings for the Task Manager, indicate the email address to send error notification.

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

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

3.3.2 Run Task On Demand

The **[Home] > [Task Manager] > [Run Task On Demand]** page 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. 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.

Links at the top of the page to help you navigate through various views of the tasks:

- **On-demand Tasks** — redirects you the [Run Task On Demand](#) page.
- **Upcoming Tasks** — redirects you the [View Upcoming Tasks](#) page.
- **Task Schedule** — redirects you the [View Task Schedule](#) page.
- **Task History** — redirects you the [View Task History](#) page.

3.3.3 View Upcoming Tasks

The **[Home] > [Task Manager] > [View Upcoming Tasks]** page lists the tasks scheduled to run within the next 24 hours. 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** to suspend it indefinitely.
 - **Yes** to suspend it now, and resume it when it is normally scheduled to run.
- **Resume** — Lets you resume a suspended task.

Links at the top of the page to help you navigate through various views of the tasks:

- **On-demand Tasks** — redirects you the [Run Task On Demand](#) page.
- **Upcoming Tasks** — redirects you the [View Upcoming Tasks](#) page.
- **Task Schedule** — redirects you the [View Task Schedule](#) page.
- **Task History** — redirects you the [View Task History](#) page.

3.3.4 View Task Schedule

The **[Home] > [Task Manager] > [View Task Schedule]** page lists all the defined tasks. You can view **Details** or **History**, as well as **Run**, a scheduled task by clicking the appropriate option:

- **Details** — Lets you edit the details of the task via the [Task Details](#).
- **History** — Displays the [Task History](#).
- **Run** — When you click **Run**, a **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.

Links at the top of the page to help you navigate through various views of the tasks:

- **On-demand Tasks** — redirects you the [Run Task On Demand](#) page.
- **Upcoming Tasks** — redirects you the [View Upcoming Tasks](#) page.
- **Task Schedule** — redirects you the [View Task Schedule](#) page.
- **Task History** — redirects you the [View Task History](#) page.

3.3.5 Task Details

The **[Home] > [Task Manager] > [View Task Schedule] > [Task Details]** page displays general information and execution details about the selected task. Clicking the label (for example, **Task Name**) displays a description in the **Details for selected item box**. You can manage the task by clicking the appropriate option:

- **Edit** — Launches the **Task Scheduler Wizard**.
- **History** — Lets you view [Task History](#) of the task.
- **Resume** — Lets you resume a suspended task.
- **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** to suspend it indefinitely.
 - **Yes** to suspend it now, and resume it when it is normally scheduled to run.
- **Delete** — Lets you delete the task from the list.

3.3.6 Task History

The **[Home] > [Task Manager] > [View Task Schedule] > [Task History]** page lists the history of tasks performed using the Task Manager. Clicking on the **Details** link displays the **[Home] > [Task Manager] > [View Task Schedule] > [Task Details]** page; for information, see [Task Details](#) in this chapter.

Links at the top of the page to help you navigate through various views of the tasks:

- **On-demand Tasks** — redirects you the [Run Task On Demand](#) page.
- **Upcoming Tasks** — redirects you the [View Upcoming Tasks](#) page.
- **Task Schedule** — redirects you the [View Task Schedule](#) page.
- **Task History** — redirects you the [View Task History](#) page.

3.3.7 View Task History

The **[Home] > [Task Manager] > [View Task History]** page lists the history of tasks performed using the Task Manager.

Links at the top of the page to help you navigate through various views of the tasks:

- **On-demand Tasks** — redirects you the [Run Task On Demand](#) page.
- **Upcoming Tasks** — redirects you the [View Upcoming Tasks](#) page.
- **Task Schedule** — redirects you the [View Task Schedule](#) page.
- **Task History** — redirects you the [View Task History](#) page.

4

Managing Caché Licensing

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

- [Configuring Caché Licensing](#)
- [Updating the License Key](#)
- [Determining License Capacity and Usage](#)
- [Identifying Users](#)

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 a `cache.key` file on every instance, except single-user 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 Configure License Servers

Configure the license servers using the System Management Portal:

1. Navigate to the **[Home] > [Licensing] > [License Server]** page.
2. This displays a list of license servers configured for this installation. From this page you can edit or delete an existing server definition or add a new server.
3. Click **Add** to configure a license server.
4. 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 UDP port number used by the license server in the **Port** box.

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

A license server is defined by the IP address of the host on which it runs and the UDP port it uses to communicate. 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:fece: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 UDP port number used at that IP address. In particular, you should ensure that the license server port number does not conflict with DCP networking port numbers used at that address.

5. Click **Edit** in the appropriate row of the license server to update. Enter the information as described in the previous step.
6. Click **Delete** to remove the license server from the configuration.

The row of the active license server is shaded when there are more than one license servers configured for this instance.

You can also view which license server is active using the **\$System.License.ShowServer** method:

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

Note: If separate instances all configure the same license server address and port, they all use the same license server. 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 Updating the License Key

Caché uses license keys to ensure proper operation of its registered sites. Caché requires a product activation key which defines the Caché features and capacity available. You may receive identifying information from InterSystems for the license key file on paper, by phone, by fax, or by computer connection. You may choose one of two options for entering license key information. You can choose to enter the license key information during installation or you can set up licensing after completing the installation. License keys are not required for single-user installations.

- You can enter license key information using the System Management Portal:
 1. Navigate to the **[Home] > [Licensing] > [License Key]** page.
 2. Click **Edit License Key** and enter the identifying information from the license that you obtained from InterSystems. The information includes the **License Capacity**, **Customer Name**, **Order Number**, **License Expiration Date** in the form mm/dd/yyyy, leaving out any leading zeroes (so that 10 July 2007 is 7/10/2007), **Authorization Key**, and **Machine ID**. Be sure to enter the information exactly as specified in the license key.

3. Click **Save**; the working version of the key is automatically updated and activated.

- You can also enter or update a license key by editing a `cache.key` file and placing it in the appropriate manager's directory on the Caché server (*install-dir/Mgr*).

If you are upgrading the license key file manually by editing the `cache.key` file, you can activate it from the System Management Portal by clicking **Activate License Key** from the **[Home] > [Licensing]** page.

Alternatively, you can invoke the `$$System.License.Upgrade` method from the `%SYS` namespace:

```
ZNspace "%SYS"
Set st=$System.License.Upgrade()
Write:st=1 !,"License key successfully upgraded"
Write:st'=1 !,"License key upgrade: ",st
```

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. Most license upgrades are expected to increase the number of license units or change from single-server to multiserver licenses. License upgrades which reduce the number of license units do not take effect until you restart Caché.

Another constraint is the amount of memory the license upgrade consumes from the generic memory heap (*gmheap* or *gmheap*) space. If *gmheap* space is not available, the number of license table entries cannot be expanded. There is also a limit to the amount of *gmheap* space (10 pages) that a license upgrade consumes. This limit permits a maximum of slightly more than 4,100 additional license units to be activated. This is the incremental count, not the total, so if you have a 30,000 Concurrent User key, you can upgrade it to slightly more than 34,100 license units without restarting the instance. If you upgrade your license by purchasing more than 4,100 additional license units, approximately 4,100 additional license units are activated, but a restart is necessary to activate the rest. You can increase the size of the [gmheap](#) (*gmheap*) setting from the **[Home] > [Configuration] > [Advanced Memory Settings]** page of the System Management Portal.

Note: If Caché cannot load all the new license units during a license key upgrade because of insufficient *gmheap* space, it logs an error message (severity = 2) in the `cconsole.log`.

4.2.1 License Troubleshooting

If, after entering your license and restarting Caché, only one user can log in, check that you have typed in the license correctly. The **[Home] > [License Usage] > [License Usage by Process]** page of the System Management Portal shows how many processes are running. You can also use the portal to display license information from the **[Home] > [Licensing] > [License Key]** page. If the key is not valid, the **CustomerName** field contains an explanation.

You can also check the license error messages in the `cconsole.log` file. This file is stored in the system manager's directory and can be viewed in the System Management Portal from the **[Home] > [System Logs] > [View Console Log]** page.

\$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, enter the following from a DOS prompt in the Bin subdirectory to get one additional terminal session for license troubleshooting purposes:

```
css cterminal <instname> -B
```

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 *Caché Class Reference* for details.

The following table shows the queries that correspond to the **[Home] > [License Usage]** pages of the System Management Portal.

Portal Page	License Query
License Activity Summary	Summary() —returns license usage summary as displayed by \$System.License.ShowSummary
License Usage by Process	ProcessList() —returns license use by the operating system process identifier (PID) as displayed by \$System.License.DumpLocalPID
License Usage by User	UserList() —returns license use by User ID

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( )
```

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 acknowledgement to update messages from each instance.

This negative acknowledgement 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 acknowledgements 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 a maximum limit of processes per user. The `$System.License.MaxConnections()` method returns the maximum value (*maxconn*) for the current implementation.

```
Set maxconn=$System.License.MaxConnections()  
Write "Maximum connections = ",maxconn
```

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 presumes that if more than *maxconn* connections are associated with a user ID, multiple users are accessing Caché through an intermediary, (for example, a firewall system) so additional license units are required. Therefore, when the (*maxconn* + 1) process starts, the number of license units allocated to that user ID changes from one to (*maxconn* + 1). Processes started by the **Job** command are allocated against the process limit of the user ID invoking the **Job** command.

Once the number of connections drops back under the maximum, the number of license units consumed is not set back to 1. Each connection continues to consume one license unit. You have to close all connections of a user ID before that user can start up to the maximum connections using one license unit again.

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](#)
- [CSP Connections and Other 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. When connections originate from a terminal server using the LAT protocol, the terminal server name and port name (obtained from `$ZIO`) are combined to form a user ID. Hence, a user with multiple sessions from a terminal server port is allocated a single license unit. Batch processes started by the `at` daemon on UNIX or from a batch queue on OpenVMS 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 every 1000 lines of Caché ObjectScript code.

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 CSP Connections and Other Special Cases

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.

Any SOAP request in which the user is not identified does not use a license unit because it is considered a “machine request”; 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 *Using SOAP and Web Services with Caché*.

Processes started by the user start routine (^%ZSTART or the older ^ZSTU) are another special case. The process running ^%ZSTART has no parent process. Therefore, a login is performed for the user ID, *User Startup*, before ^%ZSTART is called. Processes started by the **Job** command from ^%ZSTART have this user ID. If you prefer, call **\$System.License.Login(UserId)** from ^%ZSTART to change the user ID. This procedure means that ^%ZSTART 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.

Finally, it is important to understand that 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.

5

Using Caché on Windows

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


- [Caché Cube](#)
- [Starting Caché](#)
- [Stopping Caché](#)
- [Removing the Caché Cube from the System Tray](#)
- [Controlling Caché from the Command Prompt](#)

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*.

5.1 Caché Cube

The primary Caché interface on Microsoft Windows platforms is the Caché Cube. From the Caché Cube, you can start all of the Caché configuration and management tools. You can also invoke each Cube 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 Cube  appears in the system tray of the taskbar.

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

Caché Cube Menu

Getting Started
Start Caché [CACHE20091329]
Stop Caché
Studio
Terminal
System Management Portal
Documentation
DeepSee
Remote System Access ▶
Preferred Server [CACHE20091329] ▶
About...
Exit

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

Caché Cube 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 Caché ObjectScript routines. See Using Caché Studio for more information.

Caché Cube Command	Description
Terminal	Invokes the command line interpreter in the Caché programming environment. See the Using Caché Terminal guide for more information.
System 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 System Management Portal ” chapter for more information.
Documentation	Displays Caché online documentation.
DeepSee	Invokes the InterSystems DeepSee login page. See the Introduction to InterSystems DeepSee guide for more information.
Preferred Server [<i>server name</i>]	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.
Exit	Removes the Caché Cube 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 console.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 Cube. This starts the Caché instance using the specified configuration file. When Caché is not running, the **Caché Cube** icon appears dimmed.

If the Caché Cube 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 Cube 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 Instances](#) 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 Cube 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, and for broadcasting a warning message to users on the server. 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 Cube 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 Cube from the System Tray

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

5.4.1 Temporarily

If you click on the Cube in the system tray and choose **Exit**, the Cube stops and its icon no longer appears in the system tray. The Cube 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 Cube you wish to eliminate from the system tray.

5.5 Controlling Caché from the Command Prompt

You can also control a Caché instance from the Windows command prompt by running the `css.exe` program in the *install-dir*\Bin directory.

Command	Description
<code>css start <instname></code>	Starts the specified instance.
<code>css stop <instname></code>	Shuts down the named instance.
<code>css stopnoshut <instname></code>	Shuts down the named instance using INTNOSHUT^SHUTDOWN .
<code>css stopstart <instname></code>	Shuts down and restarts the named instance.
<code>css force <instname></code>	Forces down the named instance.
<code>css [run console cterminal] <instname></code>	Runs Caché in programmer mode with either no device, the console, or the terminal for \$Principal .
<code>css [run console cterminal] <instname> <routine></code>	Runs the named Caché routine in application mode with either no device, the console, or the terminal for \$Principal .
<code>css [run console cterminal] <instname> <routine> <namespace>¹</code>	Runs the named Caché routine in the indicated namespace in application mode with either no device, the console, or the terminal for \$Principal .

¹ 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.

To display the most current help file for running the **css** command, invoke **css help** from the Caché bin directory. For example:

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

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

6

Using Caché on UNIX, Linux, and Mac OS X

This chapter describes specific administrative procedures on UNIX, Linux, and Mac OS X.

Caché in release 5.1 and later does not run as the root user, it runs as a standard UNIX user; therefore, the internal mechanism for gaining access to protected files differs from previous releases of Caché. This chapter addresses the following topics:

- [UNIX User and Group Identifications](#)
- [Startup on UNIX](#)
- [Managing Caché](#)
- [UNIX File Permissions](#)

6.1 UNIX User and Group Identifications

Every Caché installation on a UNIX-based platform has an owner user ID, an installation management group ID, and a Caché system group ID. These correspond to the standard UNIX definition of users and groups.

- *Owner user ID* — has all privileges to all files and executables in the installation (`cachemgr` in these examples). If you install with Minimal initial security settings, `root` is the owner user ID.
- *Management group ID* — only a user in this group can manage the Caché instance (`sysmgr` in these examples)

- *Internal Caché system group ID* — a special Caché internal effective group ID, which also has all privileges to all files and executables in the installation. For maximum security, no actual users should belong to this group (`cacheusr` is the default).
- *Internal Caché user ID* — the effective user ID for processes started by the superserver and Job servers. Again, for maximum security, no actual users should have this user ID (`cacheusr` is the default).

6.2 Startup on UNIX

There are three (3) resources a Caché instance uses to control starting, stopping, and creating new processes:

1. The `cache.ids` file in the `install-dir\mgr` directory.
2. The key identifier information for the shared memory allocated for the running instance.
3. The shared memory itself.

This release of Caché uses additional resources to manage these basic resources, as well as to prevent instances from running concurrently on different nodes, or starting concurrently on a single node.

6.2.1 Shared Memory Object

The shared memory identifiers are kept in a memory-mapped shared object (an operating system internal data structure that may or may not be backed by a file in the file system). This object is not generally visible to users of the system (HP/UX is the one exception). It is referred to as the **ids shared memory object**. The purpose of this object is to hide the resource that contains the shared memory identifiers so that it is not as susceptible to accidental corruption or deletion.

6.2.2 Daemon Resource Locks

In addition, 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.

6.2.3 Cache.ids File

The `cache.ids` file contains the name of the node where Caché was started. Although 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 largely ignored during startup. In previous versions of Caché, the shared memory identifiers ([step 2](#), in the list above) were also stored in the `cache.ids` file, but this is no longer the case.

6.2.4 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 nfs shared files or dual-ported disks in a cluster). The shared object that keeps the identifiers (*ids shared memory object*) and the shared memory itself, however, are visible only to the node on which it was created (that is, the node where you started Caché).

6.2.4.1 Step 1. Check the Status of the Instance

The startup script runs **cache -cV** to find out the status of the instance. It first looks for the *ids shared memory object*:

- If the shared memory object is not found, a test is made for Daemon Resource Locks:
 - If no Daemon Resource Locks are held, startup proceeds to [Step 2](#).
 - If Daemon Resource Locks are held, the instance is assumed 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 **ctestop** or **ctestforce** to halt the running instance on the appropriate node. This deletes the `cache.ids` file and the *ids shared memory object*.

- If the shared memory object is found, a test is made to determine if is valid (that is, if it contains a valid shared memory identifier and the shared memory can be attached to the process successfully):
 - If the shared memory object is valid, the system is assumed to be up and running. This status is reported to the user. Startup halts.
 - If the shared memory object is not valid, the system is assumed to be in a failed state in which daemons are not running, but resources have not been removed. This status is reported to the user. Startup halts.

Action: The user must run **ctestforce** to clear the resource(s). This deletes the `cache.ids` file and the *ids shared memory object*. The [startup sequence](#) is repeated.

6.2.4.2 Step 2. Perform Recovery

The startup script tests for the existence of the CACHE.WIJ file in the *install-dir/mgr* directory:

- If the CACHE.WIJ file does not exist, startup proceeds to [Step 3](#).
- If the CACHE.WIJ file does exist, the startup script runs the recovery program (**cwdimj**). The recovery program checks for the existence of the *ids shared memory object*:
 - If the *ids shared memory object* exists, the system is assumed to be up and running. The recovery program exits with an error. Startup halts.

Action: The user must run **cstop** or **cforce** to halt the running instance.
 - If the *ids shared memory object* does not exist, the recovery program checks the Daemon Resource Locks:
 - If no Daemon Resource Locks are held, recovery assumes Caché is not running. The recovery process locks the Control Process Resource Lock to prevent concurrent startups, and performs recovery on the instance. When the process exits, the lock is released, and startup continues with [Step 3](#).
 - If Daemon Resource Locks are held, one or more daemons are running on some node for this instance. The recovery program reports this and exits with an error. Startup halts.

Action: The user must run **cstop** or **cforce** to halt the running instance on the appropriate node. This deletes the cache.ids file and the *ids shared memory object*.

6.2.4.3 Step 3. 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. At the end of the first phase (Control process startup and daemon initialization), shared memory, the *ids shared memory object*, and the cache.ids file are created.

6.3 Managing Caché

From the shell, a user with any user ID in the `sysmgr` group can run **ccontrol**. When **ccontrol** executes, it finds the proper file (`cstop`, for example) in the *install-dir/bin* directory.

The `cstart`, `cstop`, and `cforce` script files link to the `cmgr` executable, which runs in its current directory switching the user ID to `cachemgr`.

The `cmgr` executable calls these scripts, which execute the appropriate Caché executables to run Caché in the right mode for shutdown or force. 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** (super user) 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](#) for more options and information.

From the shell, a user with any user ID in the *sysmgr* group can run **ccontrol**. When **ccontrol** executes, it finds the *cstart* file in the *install-dir/bin* directory. The *cstart* script file is a link to the *cmgr* executable, which runs in its current directory switching the user ID to *cachemgr*.

The *cmgr* executable calls the *cstart* script, which executes the following:

```
cuxs -s . -c c -C...
```

This starts Caché, switching to the *cacheusr* group ID.

The Control process (CP), Write daemon (WD), Slave Write daemon (SWD), Garbage Collector (GC), Expansion daemon (ED), and Journal daemon (JD) now all run as *cachemgr:cacheusr* where *cachemgr* is the owner user ID in these examples. The Signal daemon (*csigd*) is the only Caché executable that runs as root (*root:cacheusr*). The Signal daemon is used to notify Caché processes about mail messages, and to send terminate signals to processes during shutdown.

Next, the script executes the following:

```
cuxsusr args ^STU
```

This switches to the user ID *cacheusr* and runs the startup routine.

This process creates the superserver (*cacheusr:cacheusr*) and Job servers (*cacheusr:cacheusr*). Following startup, any other daemons such as ECP and the Expansion daemon are forked from the Control process and run as *cachemgr:cacheusr*.

Only processes started by the superserver can use Job servers, since the user ID of a process must be *cacheusr* to use a Job server.

All user processes are now forked directly from the parent process, not the Control process, as has previously been the case.

You can also start, stop, or restart Caché on Linux platforms using the standard Linux methods. For example, to start Caché on SUSE Linux:

```
/etc/init.d/cache start
```

From Red Hat Linux:

```
service cache start
```

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**, which executes `cuxsession` in the `install-dir/bin` directory.

The `cuxsession` executable switches the group ID to `cacheusr` and leaves the calling user ID in place (`anyuser:cacheusr`). Running with `cacheusr` as the group ID, this executable has access to all Caché privileges and files in that installation.

Now 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.

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** (super user) 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** command:

```
ccontrol stop <instname>
```

where *instname* is the name of the Caché instance you want to stop. See [Controlling Caché Instances](#) for more options and information.

- 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/tty06
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.

6.4 UNIX File Permissions

This sample shows the permissions for files and directories where the owner user ID is *cachemgr* and the group ID chosen to start and stop Caché is *sysmgr*. The default Caché system group ID used internally is *cacheusr*

- Owner user ID is *cachemgr*
- Management group ID is *sysmgr*
- Internal Caché system group ID is *cacheusr*

Resource	Permissions	Owner	Group
-----	-----	-----	-----
Caché Registry	-rw-r--r--	Root	Other
Directories			
=====			
cachesys	-rwxrwxr-x	cachemgr	cacheusr
cachesys/mgr	-rwxrwxr-x	cachemgr	cacheusr
cachesys/bin	-rwxr-xr-x	cachemgr	cacheusr

```

in cachesys/bin
cache          -r-xr-xr-x    root    cacheusr
call_os_backup -r-xr-xr-x    cachemgr cacheusr
cbackup ->cmgr  lrwxrwxrwx    cachemgr cacheusr
cbackups       -rwx--x---    cachemgr sysmgr
ccontrol       -r-xr-xr-x    cachemgr cacheusr
cfg_upgrade    -r-xr-xr-x    cachemgr cacheusr
cforce -> cmgr  lrwxrwxrwx    cachemgr cacheusr
cforces        -rwx--x---    cachemgr sysmgr
clmanager      -rwx--s--x    cachemgr cacheusr
cmgr           -rws--x---    cachemgr sysmgr
csession       -r-xr-xr-x    cachemgr cacheusr
csigd          -rws--x---    root    cacheusr
cstart -> cmgr  lrwxrwxrwx    cachemgr cacheusr
cstarts        -rwx--x---    cachemgr sysmgr
cstat          -r-xr-x---    cachemgr sysmgr
cstop -> cmgr  lrwxrwxrwx    cachemgr cacheusr
cstops         -rwx--x---    cachemgr sysmgr
cuxs           -rwx--s--x    cachemgr cacheusr
cuxsession     -rwx--s--x    cachemgr cacheusr
cuxsusr        -rws--s--x    cacheusr cacheusr
cvendian       -rwx--s--x    cachemgr cacheusr
cvlocale       -rwx--s--x    cachemgr cacheusr
cvtcfg         -rwx--s--x    cachemgr cacheusr
cwdimj         -rwx--s--x    cachemgr cacheusr
*.so           -r-xr-xr-x    cachemgr cacheusr
licentry       -r-xr-xr--    cachemgr sysmgr
RegModule      -r-xr-xr-x    cachemgr cacheusr
wrapcmd        -r-xr-xr--    cachemgr sysmgr

in /mgr
cconsole.log   rw-rw-r--    cachemgr cacheusr

All others     rw-rw----    cachemgr cacheusr

```

Databases

The following permissions exist on databases:

```

dataset directory    cachemgr    cacheusr    rwxrwxrwx
CACHE.DAT            cachemgr    cacheusr    rw-rw----

```

Caché does not generally create user-defined database directories. (The System Management Portal allows you to do so, but this is not required.) When created by the portal, the protection and ownership is the same as shown above for database directories.

Creating Database Directories

Scripts or programs that create database directories outside of Caché must be aware that certain permissions are required on the directory. This release of Caché uses a standard group ID of `cacheusr` (this can be changed during installation). All database directories must have `rwx` permission for group, and must be owned by group `cacheusr`.

For example, if the owner user ID for an installation is `root`, then the standard `USER` database directory in `cachedb/mgr` has permissions:

```
drwxrwxr-x    2 root    cacheusr    4096 Apr 28 20:45 user
```

Failure to properly set the directory permissions may result in messages such as the following: “22:14:41 Cannot remove database lock /cachedb/data/mydirectory/cache.lck: Delete failed.”

In the following case, a script running as root creates the directory /cachedb/data/mydirectory/, and has permissions:

```
drwxr-xr-x    2 root    root          4096 May  6 22:13 mydirectory
```

Important: Caché does not interfere with the permission settings of the operating system. You can manage the permissions of files you create by adjusting your user account **umasks**.

7

Using Caché on OpenVMS

This chapter describes the following procedures on OpenVMS:

- [Starting Caché](#)
- [Stopping Caché](#)
- [Accessing the System Management Portal](#)

7.1 Starting Caché

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 OpenVMS, one of the following must be true:

- You are logged into OpenVMS as the system manager.
- You have access to CMKRNL, WORLD, BYPASS, SYSLOCK, ALTPRI, and OPER privileges.

Run the Caché startup procedure. The distribution includes a template of this startup command procedure, CACHE\$STARTUP.COM; modify this file as necessary for your site.

Run CACHE\$STARTUP.COM at the operating system level when you restart OpenVMS, as well as when you install or upgrade Caché.

To start Caché, modify the CACHE\$STARTUP.COM file to automatically start the desired Caché instance when OpenVMS starts (as described in the header text of the file.)

You can also start Caché from the command line, using the **ccontrol** command as follows:

```
ccontrol start <instname>
```

where *instname* is the Caché instance name that you chose during the installation. See [Controlling Caché Instances](#) for more options and information.

7.2 Stopping Caché

To stop Caché on OpenVMS, the requirements are the same for starting Caché. You must have access to CMKRNL, WORLD, BYPASS, SYSLCK, ALTPRI, and OPER privileges. You should have these privileges if you are logged on OpenVMS as the system manager. To stop Caché:

1. Use the **ccontrol** command:

```
ccontrol stop <instname>
```

where *instname* is the name of the Caché instance that you want to stop. See [Controlling Caché Instances](#) for more options and information.

2. This procedure displays a status report. Check for active processes in the report.
3. If there are active users on the system, you should send them a warning message that you are about to shut down the Caché system. Enter the message you want to send, and the terminal to which you want to send it. To send a message to several terminals, specify each terminal individually. For instance:

```
Do you want to broadcast a message to anyone? No=>Yes
Send a message to other terminals. Message => Please sign off
Terminal => _NTY5:
Terminal => <RETURN>
Message =>
```

4. When the system asks if you would like to see another system status, enter **Yes** to see one, or press **RETURN** if you do not want another report.
5. Confirm that you want to halt by answering **Yes**. If you answer **No**, the shutdown procedure quits, and Caché continues running.

7.3 Accessing the System Management Portal

The System Management Portal is a CSP application provided with the installation to manage a Caché instance. Unlike on other platforms, Caché does not install a private Web server on an OpenVMS instance. To use the System Management Portal to manage a Caché server on OpenVMS, you must install the CSP Gateway on a [supported Web server machine](#) and configure it to control the Caché instance on OpenVMS.

You can do this:

- **Locally** — You can download the Hewlett Packard Secure Web Server (HPSWS) for VMS. HPSWS is based on the Apache web server and can be downloaded for free from Hewlett Packard.
- **Remotely** — You can install the CSP Gateway on a remote system that is running a supported Web server. The version of Caché (specifically, the CSP Gateway components) on the remote Web server host must be from the latest Caché release. Ideally, it must be the same or a later version of the Caché system it manages.

You can access the System Management Portal through any supported Web server and operating system combination. The first of the following sections describes the procedure for setting up the CSP Gateway and Web server environment on a Windows host; the second directs you to sections of the documentation that provide information on more advanced Web server configurations.

- [Configuring the Portal on Windows](#)
- [Advanced Configuration](#)

7.3.1 Configuring the System Management Portal on Windows

This section describes how to access the Caché instance on an OpenVMS system from a remote Web server on Windows:

1. Verify that you have a supported Web server installed and running on the Windows machine.
2. On the Web server machine, install a minimal Caché shell. . During the installation process, follow these instructions below. (The detailed installation procedure is described in the “[Installing Caché on Windows](#)” chapter of the *Caché Installation Guide*.)
 - In the **Setup Type** dialog box, select **Custom** and click **Next**.
 - In the **Select Components** dialog box, clear all the components except **Web Server Gateway (CSP)** and the Web server you are configuring and click **Next**.

This creates the CSP directory structure on the Web server and creates virtual directory references for the /CSP and /CSP/Bin files in IIS or Apache. It overlays any previous configuration changes you may have made.

3. Once you have installed this minimal Caché shell on a Windows Web server machine, you must now configure the CSP Gateway to access the OpenVMS-based Caché server. On the Web server machine, navigate to the CSP Gateway Configuration page by entering the URL in a browser:

<http://localhost/csp/bin/Systems/Module.cxw> (bookmarking this URL is helpful)

This does not point to the Caché private Apache Web server; it points to a supported Web server on your system using the default port of 80.

4. Click **Server Access** and **Add Server** to create a new server configuration that connects to the remote Caché server. For this new server configuration, fill in the **Server Name**, **IP Address**, and **TCP Port** fields with the name, IP address, and superserver TCP port of the OpenVMS Caché server. Click **Save Configuration**. (You can use this Web server to support multiple Caché instances by repeating this step to create server configurations for other Caché instances.)
5. Click **Application Access** and **Add Application** to create a configuration for the path to the portal application, /csp/sys. For this path configuration, define the **Default Server** to be the **Server Name** you entered in the previous step and click **Save Configuration**.
6. Navigate to the System Management Portal using the following url:
<http://localhost/csp/sys/UtilHome.csp>, where you can now manage your OpenVMS instance.

7.3.2 Advanced Configuration

The previous section describes a configuration that uses your Web server to control one instance of Caché on an OpenVMS machine. More than likely, your environment is more complex or you need to use your Web server for other applications as well. See the following topics for details:

- To properly use other CSP applications and Caché features, configure the additional settings described in the *Advanced Web Server Configuration* section of the “[Connecting to Remote Servers](#)” chapter of this guide.
- To configure your Web server to manage one or more remote instances of Caché without interfering with your current Web server configuration, see the “[Using Caché Server Pages with a Remote Web Server](#)” chapter of the *CSP Gateway Configuration Guide*.

8

Connecting to Remote Servers

You can control remote instances from a Telnet session, from the **Remote System Access** submenu on the Caché Cube 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 Cube and point to **Remote System Access**.
3. Point to a cube 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 Cube 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 Mac OS X platform you can also use **SSH** from a command prompt on Windows to connect to your OS X 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 is a common setup to manage an OpenVMS Caché instance using the System Management Portal, which itself is a CSP application.

When connecting to the **System 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.

8.1 Define a Remote Server Connection

To use the Caché Cube 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 Cube 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 Cube menu, point to **Preferred Server**, and click **Add/Edit**.
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.

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 Cube.
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.

Input Field	Description
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.

8.1.1 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.

For example, to use a Web server on an OpenVMS instance, you need to install the Web server and the CSP Gateway on a separate machine; enter the IP address of that Web server and enter the IP address of the OpenVMS Caché instance in the first IP address field.

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 System Management Portal on the cube 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.*](#)

8.2 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 System Management Portal to the remote Caché server instance and configure the following settings from the **[Home] > [Configuration] > [Startup Settings]** page of the portal:

- **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 information on more complex configurations using CSP and remote Web servers, see the “[Using Caché Server Pages with a Remote Web Server](#)” chapter of the *CSP Gateway Configuration Guide*.

9

Using Multiple Instances of Caché

You are not limited to running only one Caché system on a machine. You can install multiple Caché systems; each has its own Caché Cube on a Windows system. These environments are referred to as Caché instances. An *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](#)

9.1 Caché Instances

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

As you install each instance of Caché you give it an instance name. To perform procedures on a particular instance of Caché, use this name. You can start, stop, and maintain each of these instances independently. You can have multiple Caché Cubes, one for each instance you have installed on your machine. All of the cube utility functions operate within the context of the instance associated with that cube.

There is a limit of fifty Caché instances that may be installed on a single machine.

To control a Caché instance from the command line, use the instance name. The following table shows the most common commands to manipulate Caché instances.

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>

* Neither available nor necessary on Windows platforms.

From the Caché Cube, 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 on “[Connecting to Remote Servers](#)” for more detailed information.

9.2 Connecting to a Caché Instance

Once you have started Caché, run and log on to the Caché shell using the **csession** command:

```
csession <instname>
```

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

Caché csession Command and Arguments

Command	Description
<code>csession <i>instname</i> -"B"</code>	Provides system administrator login to perform system tasks such as setting up licensing.
<code>csession <i>instname</i> -"U" "namespace"</code>	Specifies the login namespace.
<code>csession <i>instname</i> "[label[+offset]]^routine"</code>	Runs a routine in user mode.

To use application mode or the Caché command line on a remote server use a Telnet client, LAT terminal, or the Caché Terminal.

9.3 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]
```

In the following table, *instname* is the name of the instance that you are managing.

The optional arguments (indicated by [*arguments*]) available 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)

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

Caché ccontrol Command and Functions

Command	Description
ccontrol start <i>instname</i> [args]	Brings up an instance.
ccontrol start <i>instname</i> [<full path and file name of .cpf file>]	Starts Caché instance <i>instname</i> , optionally using the specified .cpf file. If a new .cpf file is specified, it becomes the default startup .cpf file.
ccontrol stop <i>instname</i> [args]	Shuts down an instance.
ccontrol force <i>instname</i>	Forces an instance down.

Command	Description
ccontrol list	<p>Lists all installed instances, including each instance name, installation directory, version ID, SuperServer port number, WebServer port number, Caché parameter file name, and the status of each instance, as follows:³</p> <ul style="list-style-type: none"> • crashed • down • in use by node <i>nodename</i> • inaccessible • indeterminate • mismatched executables • running • sign-on inhibited
ccontrol all	<p>Shows compact summary of all instances, including each instance name, installation directory, version ID, SuperServer port number, and the status of each instance, as follows:¹</p> <ul style="list-style-type: none"> • <blank> (inaccessible, indeterminate, sign-on inhibited, or mismatched executables) • dn (down or has crashed) • up (running) • xt (in use by another node)
ccontrol stat <i>instname</i> [args]	Retrieves system statistics.
ccontrol help	Displays most recent information about the ccontrol command. ²
ccontrol help [start stop force]	Gives function-specific help for the start, stop, and force functions.
ccontrol load <i>instname</i>	Runs the procedure for loading images (OpenVMS only).

Command	Description
<code>ccontrol unload <i>instname</i></code>	Runs the procedure for unloading images (OpenVMS only).
<code>ccontrol rename <i>instname</i> {<i>newname</i>}</code>	Renames the instance (not available on Windows systems).

Table Notes

¹ On OpenVMS and UNIX-based systems, the **all** function displays the information for each instance on a single 80-column line with no line wrapping. It truncates any field that exceeds the allowed width and suffixes the tilde (~) symbol. If you require complete information, such as for parsing or reporting purposes, use **ccontrol list**.

² To display the most current help file for running the **ccontrol** command on Windows, invoke **ccontrol help** from the Caché 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.

On UNIX-based and OpenVMS platforms, you can run the **ccontrol help** command from any location.

³ On OpenVMS-based systems, you may need to use a privileged account (for example, SYSTEM), or give a non-privileged account READALL, BYPASS, or SYSPRIV privileges, when using the **ccontrol list** function to display meaningful status information about installed instances.

You may have your own procedures that involve using the scripts **cstart** and **cstop**, among others. Caché can accommodate these procedures. In fact, **ccontrol** invokes these scripts.

For an example of creating user start and stop procedures, see the [Use of the ^%ZSTART and ^%ZSTOP Caché Routines](#) article.

Important: The **ccontrol** command has several more functions. The help function displays available functions; others are internal to InterSystems and do not appear in the help display.

9.4 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).
- Only one instance per NIC (network interface card) can use DDP (Distributed Data Protocol) or DCP (Distributed Cache Protocol) over Ethernet. DDP and DCP are older distributed data protocols that are supported for interoperability with legacy applications. If a machine has more than one NIC, each instance must point to a unique NIC. This restriction does not apply to DCP over UDP (User Datagram Protocol).
- 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.

9.4.1 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 **[Home] > [Configuration] > [Memory and Startup]** page of the System Management Portal, and you can change the Web server and Telnet port values after installation from the **[Home] > [Configuration] > [Startup Settings]**

and **[Home] > [Configuration] > [Device Settings] > [Telnet Settings]** pages, respectively, of the System 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 Cube tools on multiple instances from one Windows client.

10

Achieving High Availability

This chapter provides an overview of the mechanisms that Caché provides to maintain a highly available and reliable system. It describes strategies for recovering quickly from system failures while maintaining the integrity of your data.

This chapter discusses the following topics:

- [Surviving a Crash](#)
- [Write Image Journaling](#)
- [Backup Strategy](#)
- [Logical Data Protection](#)
- [Shadow Journaling](#)
- [System Failover Strategies](#)

Most of these subjects are described in further detail in the *Caché Data Integrity Guide* and the *Caché High Availability Guide*.

10.1 Surviving a Crash

Both internal and physical integrity can be threatened by hardware failures caused by malfunctions or external events such as fire or loss of electric power. However, modern hardware helps avoid system crashes, and Caché contains safeguards and offers strategies that allow you to recover data as quickly and completely as possible.

Two mechanisms critical to the recovery process are write image journaling and establishing a consistent transaction journaling and backup strategy.

Caché write image journaling technology protects against internal integrity failures due to system crashes. Caché backup and journaling systems provide rapid recovery from physical integrity failures. These utilities can be used while applications are running and the database is being updated. Logical database integrity is ensured through transaction processing statements and automatic rollback.

Important: InterSystems recommends isolating all WIJ (cache.wij), PIJ (cache.pij, cache.pijxxx), and journal (yyyyymmdd.nnn) files from the database (cache.dat and cache.ext) files. Placing these files on a separate disk may be critical if there is a crash and the database is corrupted. It minimizes the risk of these files also being corrupted, increasing the opportunity for successful recovery.

10.1.1 Database Integrity Protection

To recover databases after a loss of structural integrity, restore the backup and then apply the subsequent database changes from the journal.

This Caché recovery process is designed to provide maximal protection:

- It uses the “roll forward” approach. If a system crash occurs, the recovery mechanism completes the updates that were in progress. By contrast, other systems employ a “roll back” approach, undoing updates to recover. While both approaches protect internal integrity, the roll forward approach used by Caché does so with reduced data loss.
- Caché protects the sequence of updates: if an update is present in the database following recovery, preceding updates are also present. Other systems which do not correctly preserve update sequence may yield a database that is internally consistent but logically invalid.
- The incremental backup file structures are protected, as well as the database. A valid incremental backup can thus be performed following recovery from a crash.

10.2 Write Image Journaling

Write image journaling maintains the internal integrity of the Caché database, and is the foundation of the database recovery process. Write image journaling is automatically enabled when you start Caché, and works for all Caché directories.

Rather than writing directly from memory to the database, the Caché Write daemon system process (**WRTDMN**) uses an intermediate file, the write image journal. This file is usually named cache.wij.

Write image journaling technology uses a two-phase process of writing to the database. When Caché starts, it automatically checks the write image journal and runs a recovery procedure if it detects that an abnormal shutdown occurred. When Caché indicates successful completion, the internal integrity of the database is restored.

For more detailed information on the two-phase write protocol and the recovery procedure, see the [“Write Image Journaling”](#) chapter of the *Caché Data Integrity Guide*.

10.3 Backup Strategy

Selecting, implementing, and validating an appropriate backup plan are critical aspects of proper Caché administration and maintaining a highly available system. Caché contains many built-in tools to help implement a backup strategy. Your backup procedures can use internal Caché tools and utilities, be completely external to Caché, or be a mix-and-match approach using both internal and external tools. You can vary these strategies based on your environment: the operating system, preferred backup utilities, disk configurations, and backup devices.

All Caché content to back up is in `cache.dat` and their corresponding `cache.ext` files. In addition, you should also back up the transaction log—the journal files—nightly to ensure you have the capability to restore the transactional integrity of your database by rolling back uncommitted transactions (the databases may have contained partial transactions at the time of the backup).

The best and most common strategy for backing up Caché databases is to perform daily online concurrent backups during the slowest period of a production day. You do not have to stop Caché to back up your data; therefore, you do not impact users of the system. There are various ways to backup up a Caché system.

For detailed information on the importance of backups and journals, types of backup methods, and procedures for configuring backup tasks and running backups and restores, see the [“Backup and Restore”](#) chapter of the *Caché Data Integrity Guide*.

10.4 Logical Data Protection

Applications need to ensure that sets of related database changes work correctly. This type of application integrity is called logical integrity. Caché protects the logical integrity of databases using the following:

- [Transaction Processing](#)
- [Transactions and Locking](#)
- [Global Journaling](#)

For more information, see the [“Transaction Processing”](#) chapter of *Using Caché ObjectScript* and the [“Managing Transactions”](#) section in *Using Caché Globals*.

10.4.1 Transaction Processing

A set of related changes is called a “transaction.” Examples include updating a patient's medical record, transferring funds from one bank account to another, and reserving an airline seat.

In general, you must define the scope of a transaction although Caché can infer the boundaries in some cases. For example, suppose that the computer malfunctions during a transaction. The two-phase write protocol protects the internal integrity of the database, but not the logical or application-level integrity. It does not prevent you from ending up with half of a patient record or two-thirds of airline seat processing.

The best protection is offered by transaction processing. In a Caché application, the start of a unit of work is indicated with a **TStart** command. If the transaction completes normally, a **TCommit** command is issued. If a transaction does not complete successfully, Caché performs a process called *rollback*, in which all values altered by the transaction are restored to their prior state. When a transaction is rolled back, it is as if the failed transaction never occurred.

Three situations may trigger a rollback:

- The application itself may detect a problem and request a rollback. Often this is done from an application error-handling routine following an error. You can use Caché ObjectScript to implement these techniques in your application, or rely on automatic transaction processing in Caché SQL.
- If an operator terminates a process, Caché automatically rolls back incomplete transactions.
- Following an abnormal termination of the entire Caché system, as in the case of a hardware failure, rollback of all incomplete transactions occurs the next time Caché starts.

For rollback to occur, Caché stores additional information in the journal.

Let's suppose that a transaction involves the transfer of \$100 from a savings account with a starting balance of \$1,000, to a checking account with a balance of \$500. Without transaction processing, there are two journal entries:

```
SET SAVINGS 900
SET CHECKING 600
```

Now suppose that transaction processing is invoked by preceding these updates with a command to begin the transaction, and following them with a command to commit the transaction. In this case, the journal *conceptually* contains:

```
START
SET SAVINGS 900
OLD SAVINGS 1000
SET CHECKING 600
OLD CHECKING 500
COMMIT
```

In addition to the new values of 900 and 600, the journal contains the pre-transaction values of 1000 and 500. In the event that a rollback is required before the transaction is committed, these old values are used.

When a system crash occurs, rollback happens automatically. First, recovery via the two-phase write protocol restores the internal consistency of the database. Next, all incomplete transactions (those that began but were never committed) are rolled back.

10.4.2 Transactions and Locking

You use the **Lock** command in Caché applications to protect against simultaneous access to data.

For example, an application might contain the following steps:

```
TSTART                                //begin transaction
  Lock  +^SAVINGS (ACCOUNT, "BALANCE" )    //lock record
  Set   ^SAVINGS (ACCOUNT, "BALANCE" ) = ^SAVINGS (ACCOUNT, "BALANCE" ) - 100
  Lock  -^SAVINGS (ACCOUNT, "BALANCE" )    //unlock record
  Lock  +^CHECKING (ACCOUNT, "BALANCE" )    //lock record
  Set   ^CHECKING (ACCOUNT, "BALANCE" ) = ^CHECKING (ACCOUNT, "BALANCE" ) + 100
  Lock  -^CHECKING (ACCOUNT, "BALANCE" )    //unlock record
TCOMMIT
```

The value of *SAVINGS* becomes unavailable to other processes when it is locked, and available again as soon as it is unlocked.

Now suppose that a transaction fails as the application is updating *CHECKING*. Transaction rollback occurs, restoring *SAVINGS* to its pre-transaction value. But, what if between the instant the application unlocks *SAVINGS* and transaction rollback completes, another process accesses *SAVINGS*? If it examines this value, it may carry out further processing that is erroneous. Worse yet, if it changes this value, rollback cannot occur.

Caché automatically avoids this problem by deferring any unlocks within a transaction until the transaction is committed or rolled back. Within the transaction, the data appears to be unlocked, permitting a subsequent lock of the same value in the same transaction to work correctly. Outside the transaction, however, the data remains locked.

10.4.3 Global Journaling

Global journaling preserves changes in the database since the last backup. While a backup is the cornerstone of physical recovery, it is not the complete answer. Restoring the database from a backup makes the contents of the database current as of the time the backup was completed. Typically, this is a long time (at least a number of hours) before the point at which physical integrity was lost. What happens to all the database changes that occurred since then? The answer lies with journaling.

Each computer running Caché keeps a journal. The journal is a file that keeps a time-sequenced log of changes that have been made to the database since the last backup. The process is redundant and

logical and does not use the Caché Write daemon. You can turn on journaling for all modifications to a database, or turn it off for a database.

The default setting when you create a new database is to have journaling on. At installation, by default, the CACHEAUDIT, CACEHLIB, DOCBOOK, and SAMPLES databases are not journaled. You can, however, turn journaling on for these databases. The CACHETEMP database is never journaled.

Selective database journaling avoids the overhead of saving updates to databases for which recovery is not needed. Caché transaction processing works with journaling to maintain the logical integrity of data. The journal also contains **Set** and **Kill** operations for all globals in databases marked for journaling.

When Caché starts, all journal entries since the last Write daemon pass are reapplied. Since user processes update the journal concurrently, rather than through the Write daemon, this approach provides added assurance that updates prior to a crash are preserved.

Backups and journaling are daily operations that allow you to recreate your database. If a failure renders your database inaccessible or unusable, you can restore the backups and apply the changes in the journal to recreate your database.

For more detailed information, see the “[Journaling](#)” chapter of the *Caché Data Integrity Guide*.

10.5 Shadow Journaling

Shadow journaling enables a secondary computer to maintain a “shadow” version of selected databases as they are updated on a primary machine. By continually transferring journal information from the primary machine to the secondary machine, shadow journaling makes it possible to fail over to a database which is within only a few transactions of the primary database.

By itself, shadow journaling does not ensure successful failover, but it is a very simple and inexpensive way to maintain a disaster recovery system. Often, this approach is also used to update a report server, where ad hoc reporting tasks can operate on current data without affecting production.

Shadow journaling monitors database activity on a primary system and causes the same activity to occur on a secondary system. It does this by means of a shadow client service running on the secondary system that continually requests journal file details from a shadow server service running on the primary system. The shadow server service responds by sending the details of the **Set** or **Kill** journal record entry to the secondary system over a TCP connection.

For more detailed information, see the “[Shadow Journaling](#)” chapter of the *Caché Data Integrity Guide*.

10.6 System Failover Strategies

Caché fits into all common high availability configurations promoted by operating system providers, including Microsoft, IBM, HP, Compaq, and EMC. Caché provides easy-to-use, often automatic, mechanisms that integrate easily with the operating system to provide high availability. There are four general approaches, in the order of increasing availability:

- *No Failover* — for applications that are most likely not business-critical.
- *Cold Failover* — a common and inexpensive approach that involves switching to a standby system in case of a failure.
- *Warm Failover* — exploits a standby system that is ready to accept users instantly after a production system failure.
- *Hot Failover* — can be complicated and expensive, but comes closest to ensuring 100% uptime.

Failover Strategy Impact

Approach	Recovery Time	Expense	User Impact
No Failover	Unpredictable	No cost to low cost	High
Cold Failover	Minutes	Moderate	Moderate
Warm Failover	Seconds	Moderate to high	Low
Hot Failover	Immediate	Moderate to high	None

Each of these approaches is described in greater detail in the “[System Failover Strategies](#)” chapter of the *Caché High Availability Guide*.

Variations on these strategies are also available. For further information to help you develop a failover and backup strategy tailored for your environment, or to review your current backup practice, please contact the [InterSystems Worldwide Response Center](#).

