



Caché System Administration Guide

Version 5.2
01 September 2006

Caché System Administration Guide
Caché Version 5.2 01 September 2006
Copyright © 2006 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.



The Caché product and its logos are registered trademarks of InterSystems Corporation.



The Ensemble product and its logos are registered trademarks of InterSystems Corporation.



The InterSystems name and logo are trademarks of InterSystems Corporation.

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.

Caché, InterSystems Caché, Caché SQL, Caché ObjectScript, Caché Object, Ensemble, InterSystems Ensemble, Ensemble Object, and Ensemble Production are trademarks of InterSystems Corporation. All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

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

Introduction	1
1 Using the System Management Portal	3
1.1 Starting the Portal	3
1.1.1 Portal Login Page	5
1.1.2 Portal CSP Application Structure	5
1.2 Portal Pages Overview	6
1.2.1 Common Portal Procedures	7
1.3 Portal Functions Overview	7
1.3.1 System Administration	8
1.3.2 Data Management	8
1.3.3 Operations	9
2 Configuring Caché	11
2.1 Configuring Data	13
2.1.1 Configuring Namespaces	13
2.1.2 Configuring Databases	16
2.2 Configuring System Information	19
2.2.1 Memory and Startup Settings	19
2.2.2 Studio Source Control Settings	20
2.3 Configuring Task Manager Settings	20
2.3.1 Schedule Tasks	20
2.3.2 Configure E-Mail Settings	20
3 Managing Caché	21
3.1 Maintaining Local Databases	21
3.1.1 Show Free Space	22
3.1.2 Check Database Integrity	24
3.1.3 View Integrity Log	25
3.2 Controlling Caché Processes	25
3.2.1 Display Process Information	26
3.2.2 Display Process Details	28
3.2.3 Display Process Variables	29
3.2.4 Broadcast Messages to Terminals	29
3.3 Using the Task Manager	30

4 Managing Caché Licensing	31
4.1 Configuring Caché Licensing	31
4.1.1 Configure License Servers	32
4.2 Maintaining License Key	33
4.2.1 License Troubleshooting	34
4.3 Determining License Capacity and Usage	35
4.4 How Caché Logins Work	37
4.4.1 Identifying Users	38
4.4.2 License Logins	38
4.4.3 CSP Connections	39
5 Using Caché on Windows	41
5.1 Caché Cube	41
5.2 Starting Caché	43
5.3 Stopping Caché	44
5.4 Controlling Caché from the Command Prompt	45
6 Using Caché on UNIX, Linux, and Mac OS X	47
6.1 UNIX User and Group Identifications	47
6.2 Managing Caché	48
6.2.1 Starting Caché	49
6.2.2 Running Caché	50
6.2.3 Stopping Caché	50
6.3 UNIX File Permissions	51
7 Using Caché on OpenVMS	55
7.1 Starting Caché	55
7.2 Stopping Caché	56
7.3 Accessing the System Management Portal	57
7.3.1 Configuring the Portal on Windows	57
7.3.2 Advanced Configuration	58
8 Connecting to Remote Servers	61
8.1 Define a Remote Server Connection	62
8.1.1 Using Optional Server Fields	64
8.1.2 Advanced Web Server Configuration	65
8.2 Using Caché Server Pages with a Remote Web Server	65
8.2.1 Install Caché Shell on Web Server Machine	66
8.2.2 Configure the CSP Gateway	66

8.2.3 Create Directories on Web Server	67
8.2.4 Add Virtual Directories to IIS	68
8.3 Accessing CSP on Multiple Caché Servers	69
8.4 Additional Resources	71
9 Using Multiple Instances of Caché	73
9.1 Caché Instances	73
9.2 Connecting to a Caché Instance	74
9.3 Controlling Caché Instances	75
9.4 Configuring Multiple Caché Instances	76
9.4.1 Set Port Numbers	77
10 Achieving High Availability	79
10.1 Surviving a Crash	79
10.2 Write Image Journaling	80
10.3 Backup Strategy	81
10.4 Logical Data Protection	82
10.4.1 Transaction Processing	82
10.4.2 Transactions and Locking	83
10.4.3 Global Journaling	84
10.5 Shadow Journaling	85
10.6 System Failover Strategies	85

List of Figures

System Management Portal Home Page	4
Sample Namespace Mapping	16
Caché Cube Menu	42

List of Tables

System Administration Tasks	8
Database Management Tasks	9
System Operation Tasks	10
System Configuration Tasks	11
Connectivity Tasks	12
Local Databases List Information	21
Local Databases Free Space Information	23
Process Management Functions	26
Process Column Information	26
Possible Process States	27
Caché Cube Commands	42
Add Connection Input Fields	62
Control Commands	74
Caché csession Command and Arguments	74
Caché ccontrol Command and Functions	75
Failover Strategy Impact	86

Introduction

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. On OpenVMS systems you must 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](#)

1

Using the System Management Portal

Caché offers different ways to perform system administration and management tasks. This release of Caché introduces the InterSystems System Management Portal, a CSP application that replaces the functions of Caché Explorer, SQL Manager, Caché Control Panel, and the Configuration Manager found on the Caché Cube in previous releases.


An advantage of this approach is that it is no longer a requirement that Caché 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 the System Management Portal icon . 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:

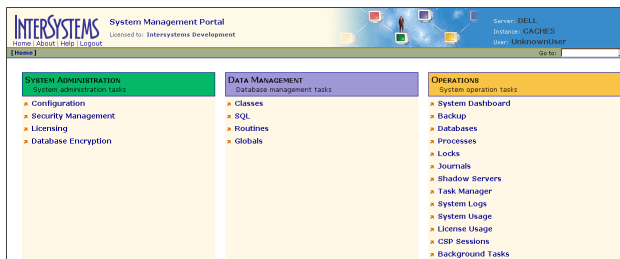
`http://machine:port/csp/sys/UtilHome.csp`

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:8972/csp/sys/UtilHome.csp>.

- 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.

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

Authenticated Access — Requires a user name and password

Only Unauthenticated Access

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), leave both the **User Name** and **Password** fields blank and click **Login**; this connects you to the portal with the `UnknownUser` account, which has `%All` privileges.

Important: If the only authentication allowed is unauthenticated for the portal CSP application, you *cannot* log in to the portal with a user name and password combination—even a valid one. If you attempt to use such a combination, the login attempt fails, most likely with the following message: `ERROR #822: Access Denied`. 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.

Authenticated Access

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.

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 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 the string.

1.3 Portal Functions Overview

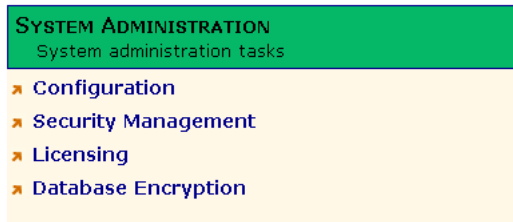
The portal is divided into three functional areas:

- **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:



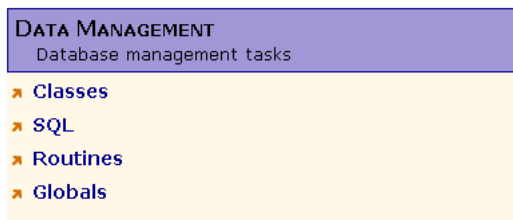
The following table displays each major subcategory and lists the most appropriate documentation source for detailed information on the topic.

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>

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 Multidimensional Storage</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 High Availability Guide</i> .
Databases	Maintaining Local Databases section of the “Managing Caché” chapter of this guide.
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 High Availability Guide</i> .
Shadow Servers	“ Shadow Journaling ” chapter of the <i>Caché High Availability 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	Viewing CSP Sessions section of the “Monitoring Caché Using the System Management Portal” chapter of the <i>Caché Monitoring Guide</i> .
Background Tasks	Viewing Background Tasks section of the “Monitoring Caché Using the System Management Portal” 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 two categories: **System Configuration** and **Connectivity**. These, in turn, are divided into subcategories. This chapter describes some of the topics; other topics have separate chapters 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
Namespaces	Configuring Namespaces section of this chapter
Local Databases	Configuring Databases section of this chapter
Remote Databases	Configuring Databases section of this chapter
Memory and Startup	Configuring System Information section of this chapter
Database Backup Settings	Configuring Caché Backup Settings section of the “Backup and Restore” chapter of the <i>Caché High Availability Guide</i>

Menu Item	Documentation Source
Journal Settings	Configuring Journal File Settings section of the “Journaling” chapter of the <i>Caché High Availability Guide</i>
Task Manager Settings	Configuring Task Manager Settings section of this chapter
SQL Settings	Caché SQL Reference
Studio Source Control Settings	Configuring System Information section of this chapter
Monitor Settings	“ Monitoring Caché Using BMC PATROL ” and “ Monitoring Caché Using SNMP ” appendixes of the <i>Caché Monitoring Guide</i> .
Advanced Settings	“ Caché Advanced Configuration Categories ” chapter of <i>Caché Advanced Configuration Settings Reference</i>

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 High Availability Guide</i>
SQL Gateway Connections	“ Configuration ” chapter of <i>Using the Caché SQL Gateway</i>
Legacy Network Connections	Consult your legacy documentation
CSP Gateway Management	“ CSP Configuration ” chapter of <i>Using Caché Server Pages</i>

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 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.

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

2.1.1 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.

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

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

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

2.1.1.1 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.

1. Enter a **Name for the namespace**.

Namespace names are at least one character long, starting with an alphabetic character or a percent sign (%), and followed by an arbitrary number of alphanumerics, 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. After entering the required information, click **Save** to add your namespace to the configuration.

2.1.1.2 Modify 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** and the **Default Database for Routines** 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.

2.1.1.3 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. Click the appropriate choice to begin mapping:

- [Global Mappings](#)
- [Routine Mappings](#)
- [Package Mappings](#)

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. Click **New Global Mapping**.
2. Select the **Global database location** database where the global is located.
3. Enter the **Global name**.
4. 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" )
($$$Text( "BEGIN" )):( "X" )
("Y"):( $$$Text( "END" ) )
```

5. Select the **Collation**.
6. Select the **Lock Database Location**.
7. Click **OK**.

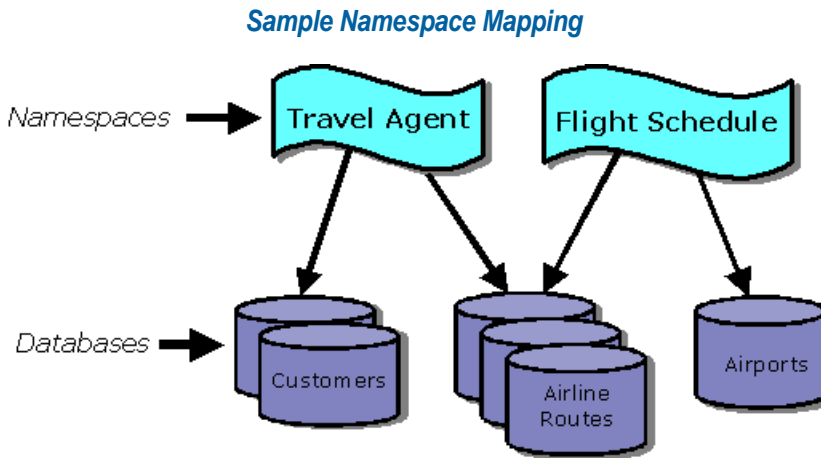
Routine Mappings

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

Package Mappings

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

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.

2.1.2 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 paragraphs:

Local Database Wizard

Remote Database Wizard

Local Database Wizard

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 a database name in the text box and click **Next**.

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.

If encryption is activated, you may encrypt this database by selecting the **Encrypt Database** check box.

3. Enter a directory name or click **Browse** to select the database directory.
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 further configure and manipulate 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.

Remote Database Wizard

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.

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.

2.2 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. You can maintain alternate versions of the configuration file, but you must name the one to activate at startup Cache.cpf.

There are a few settings you may want to review on a new installation:

- [Memory and Startup Settings](#)
- [Studio Source Control Settings](#)

There are a variety of advanced options available; however, these topics are not critical to running most Caché systems. These advanced options are described in various Caché topic-specific guides and reference books that you can access from the documentation home page.

2.2.1 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.

2. If you are running on a Microsoft Windows platform, you can set your Caché instance to start automatically when the system starts by selecting the **Start Caché on System Boot** check box.
3. You can change the **SuperServer Port Number** for this Caché instance.
4. Click **Save** to save your modifications; restart Caché to activate them.

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.

2.2.2 Studio Source Control Settings

For each namespace, you can set how and if your developers use source control.

See the `%Studio.SourceControl.ISC` class documentation in the *Caché Class Reference* for detailed information about this setting.

2.3 Configuring Task Manager Settings

From the **[Home] > [Configuration] > [Task Manager Settings]** page of the System Management Portal you can view and edit existing tasks, schedule new tasks, and configure email settings the Task Manager uses for notification. The menu contains the following items:

- [Schedule Tasks](#)
- [Configure E-Mail Settings](#)

2.3.1 Schedule Tasks

From the **[Home] > [Configuration] > [Task Manager Settings] > [Task Schedule]** page you can view, edit, and delete existing tasks and schedule new tasks:

- To change an existing task, click **Edit** or **Delete** in the appropriate row for the setting.
- To schedule a new task, click **Schedule New Task** to invoke the Task Scheduler Wizard.

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

2.3.2 Configure E-Mail Settings

From the **[Home] > [Configuration] > [Task Manager Settings] > [E-Mail Settings]** page you can configure email settings the Task Manager uses for notification.

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 and with which permissions.
Encrypted	Indicates whether the database is encrypted.
Journal	Indicates whether globals in the database are journaled with a Y or an N.
Dismount / Mount	One of two action buttons to dismount or mount the database

From this page you can perform the following database tasks:

- [Show Free Space](#)
- [Check Database Integrity](#)
- [View Integrity Log](#)

3.1.1 Show Free Space

You can check the free space available on databases from the **[Home] > [Databases]** page of the System Management Portal.

Click **Freespace** to display a page which shows a list of local databases with the following free space information:

Local Databases Free Space Information

Column Heading	Definition
Name	The database name.
Directory	The system directory in which the database resides.
Max Size	The maximum size allocated to which the database can grow, in gigabytes.
Size	The current allocated size of the database, in megabytes.
Expansion Size	
Available	
%Free	
Disk Free Space	
Status	The status of the directory, which indicates if the database is mounted and with which permissions.

Free Space Utility

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:\CacheSys\Mgr\User\
Device:
Right margin: 80 =>
```

```

                                Cache Database Free Space
                                Sep 22 2005 12:49 PM
Database                        Max Size  Size      Available %Free   Disk Free
C:\CacheSys\Mgr\User\          Unlimited 1MB      0.59MB    59      18GB
```

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:

Managing Caché

```
USER>zn "%SYS"  
%SYS>Do ^%FREECNT  
Database directory to show free space for (*=All)? *  
  
Databases Selected  
-----  
C:\CacheSys\Mgr\  
C:\CacheSys\Mgr\CacheLib\  
C:\CacheSys\Mgr\Cachetemp\  
C:\CacheSys\Mgr\Docbook\  
C:\CacheSys\Mgr\Samples\  
C:\CacheSys\Mgr\User\  
C:\CacheSys\Mgr\cacheaudit\  
Device:  
Right margin: 80 =>
```

```
Cache Database Free Space  
Sep 22 2005 12:52 PM  
Database Max Size Size Available %Free Disk Free  
C:\CacheSys\Mgr\ Unlimited 34MB 14MB 41.17 18GB  
C:\CacheSys\Mgr\CacheLib\ Unlimited 80MB 2.6MB 3.25 <- 18GB  
C:\CacheSys\Mgr\Cachetemp\ Unlimited 1MB 0.35MB 35 18GB  
C:\CacheSys\Mgr\Docbook\ Unlimited 101MB 8.8MB 8.71 18GB  
C:\CacheSys\Mgr\Samples\ Unlimited 15MB 7.8MB 52 18GB  
C:\CacheSys\Mgr\User\ Unlimited 1MB 0.59MB 59 18GB  
C:\CacheSys\Mgr\cacheaudit\ Unlimited 1MB 0.59MB 59 18GB  
%SYS>
```

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 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.
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.

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.

There are three additional entry points to this routine:

- **Do CheckPointer^Integrity** asks for a directory and a pointer block at which to start checking.
- **Do Silent^Integrity(logfilename)** starts a background job which does an integrity check on all databases and puts the output in a file specified by the *logfilename* parameter. 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 OS filename. The latter makes a copy of *logfilename*.

3.1.3 View Integrity Log

You can view the log resulting from the background job:

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	Navigate to the [Home] > [Processes] page. Click Processes under the operations column of the Home page.
Display process details	Click Details on the right hand column of the selected process on the [Home] > [Processes] page.
Stopping a process	Click Resume , Suspend , or Terminate as desired after clicking Details of the selected process.
Display process variables	Click Variables after clicking Details of the selected process.
Broadcast messages to terminals	Click Broadcast on the 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.

Column Heading	Definition
Globals	Number of global references, including updates, executed (database reads and writes) since the process entered Caché.
State	Process state. See the Possible Process States table below for an explanation of each state.
Client Name	Name of the client system that initiated 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.
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.

State	Process
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 bit strings 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

You can display more detailed information about any process from the **[Home] > [Processes]** page of 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.

Stopping a Process

From this page you can also stop or resume a process. You can stop a process in one of two 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 a Process

You may want to suspend a process if you are not sure what a process is doing and want to investigate, or if a more important process is trying to run and needs the CPU cycles. From the **Operations** column of the **[Home]** page of the System Management Portal:

1. Click the **Processes** menu item.
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. From the **Operations** column of the **[Home]** page of the System Management Portal:

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

3.2.3 Display Process Variables

3.2.4 Broadcast Messages to Terminals

You can broadcast messages to the terminals associated with a selected process, or all processes. These utilities are useful for asking people to sign off the system. However, you must

use them 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 job 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]** 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 your 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. When 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 execute tasks configured as on-demand, view upcoming task manager activities, and view a history of completed tasks.

4

Managing Caché Licensing

The license system in Caché 5.2 remains the same as that in Caché 5.1.

Important: Licenses from versions prior to Caché 5.1 do not work with Caché 5.2. Please contact the [InterSystems Worldwide Response Center](#) (WRC) for an appropriate key if you are upgrading.

The following topics are covered in this chapter:

- [Configuring Caché Licensing](#)
- [Maintaining License Key](#)
- [Determining License Capacity and Usage](#)
- [How Caché Logins Work](#)

Important: *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 system regardless of how many Caché instances run on a system. 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.

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 the IP address of the license server in the **Name** box and the UDP port number used by the license server in the **Port** box.

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*). 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 insure 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()
```

4.2 Maintaining License Key

Caché uses license keys to ensure proper operation of its registered sites. Caché requires a product activation key that 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 chose 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 or upgrade a license key by editing a `cache.key` file and placing it in the appropriate manager's directory on the Caché server (`CacheSys/Mgr` by default); or you can maintain 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**, **Expiration Date** in the form mm/dd/yyyy, leaving out any leading zeroes (so that 10 July 2005 is 7/10/2005), **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.

If you are upgrading the license key file manually by editing the `cache.key` file, invoke the **\$System.License.Upgrade** method from the `%SYS` namespace:

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

In general there is no need to restart the configuration.

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 (*GenericHeapSize* 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 7,000 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 37,000 license units without restarting the configuration. If you upgrade your license by purchasing more than 7,000 additional license units, approximately 7,000 additional license units are activated, but a restart is necessary to activate the rest. You can increase the size of the [GenericHeapSize](#) setting from the **[Home] > [Configuration] > [Advanced Settings]** page of the System Management Portal.

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 them. 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; they are described in the *Caché Class Reference*.

The %SYSTEM.License class entry in the *Caché Class Reference* describes the information displayed for the following queries that also correspond to the **[Home] > [License Usage]** pages of the System Management Portal as shown in the following table.

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 invoke 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()
```

Methods to Show Local License Information

The subroutines listed below dump the contents of license tables contained locally in configuration-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

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.

4.4 How Caché Logins Work

This section covers the following topics:

- [Identifying Users](#)
- [License Logins](#)
- [CSP Connections](#)

4.4.1 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.

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.

4.4.2 License Logins

There are two modes of license login: automatic and explicit. Automatic login remains the default in Caché 5.1. 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 explicit login is selected, no automatic detection of a user ID is attempted. The application must explicitly call the `$$System.License.Login(UserIdentifier)` method to supply the license user ID and acquire a license.

Explicit login is enabled by calling the `$$System.License.DeferUserIdentification([0 or 1])` function. This call can be made 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 login is deferred, the customer 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.

4.4.3 CSP Connections

CSP connections are a special case for logins. We strongly recommend 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.

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](#)
- [Controlling Caché from the Command Prompt](#)

5.1 Caché Cube

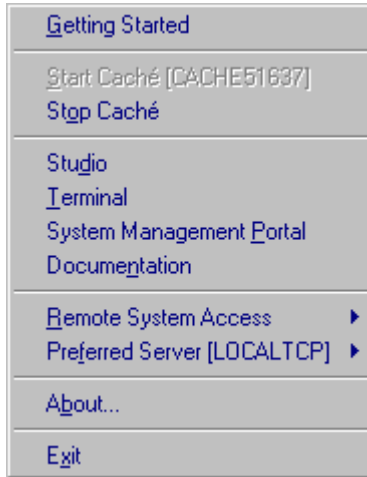
The primary Caché interface on Microsoft Windows platforms is the Caché Cube. From the Caché Cube, you can start all of the Caché configuration and management tools. You can also invoke each Cube command from a shortcut or command line.

Correspondingly, you can initiate many of the Caché tools from the Windows **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 on the Caché Cube, a menu appears with commands to use the Caché utilities and programming environments.

Caché Cube Menu



The following commands are available from the Caché Cube menu:

Caché Cube Commands

Caché Cube Command	Function
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.
Stop Caché	Shuts down or restarts the local Caché instance.
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.
Terminal	Invokes the command line interpreter in the Caché programming environment. See the “ Caché Terminal ” article for more information.

Caché Cube Command	Function
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.
Remote System Access	Displays a submenu of all the Caché utilities with access to any defined remote Caché server. The Caché Telnet command starts the Caché terminal emulator and allows you to choose your host from within the Telnet window. See the “ Connecting to Remote Servers ” chapter for more information.
Preferred Server [Server name]	Shows a list of remote servers and maintains server connections by using the Add/Edit command on the submenu. The preferred server appears in brackets and has a check mark next to it in the server list. See the Define a Remote Server Connection section of the “Connecting to Remote Servers” chapter for more information.
About	Displays Caché version and build information.
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

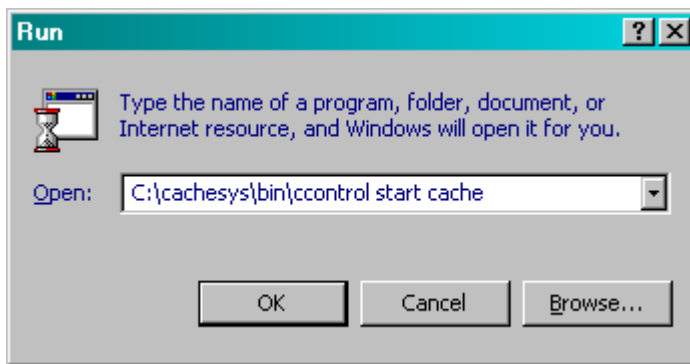
To start Caché, run the startup procedure at the system level. This procedure runs using either the default configuration file or a configuration file you specify.

If you have any trouble starting Caché, view the console log using the System Management Portal 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, access the Bin folder in the Caché instance directory (CacheSys by default) and double-click the csystray.exe file.

Alternatively, you can enter these commands from the <cache-install-dir>/bin directory in the **Open** box of the **Run** item on the Windows **Start** menu. For example, to start the instance named cache, enter the following command and click **OK**.



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. InterSystems recommends that you use **Stop Caché** to shut down Caché to ensure that it closes properly.

Note: You cannot cancel a shutdown once the countdown reaches 0 and the shutdown procedures have started.

5.4 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 `<cache-install-dir>\Bin` directory (CacheSys\Bin by default).

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 instance bin directory. For example:

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

The command displays the `CssHelp.html` file in your browser; the file is in the `<cache-install-dir>\Help` directory (CacheSys\Help by default).

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](#)
- [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.

The examples in this chapter assume that you install Caché with following values:

- Owner user ID is `cachemgr`
- Management group ID is `sysmgr`
- Default Caché system group ID is `cacheusr`

Only a user in the `sysmgr` group can manage the Caché instance. The owner user (`cachemgr` in this chapter) has all privileges to all files and executables in the installation. The `cacheusr` group is a special Caché internal 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.

There is also a special Caché internal user ID called `cacheusr`. This is the generic user ID for processes started by the SuperServer and Job servers. Again, for maximum security, no actual users should have this user ID.

The default Caché system user in the installation is the owner user ID: root if using Minimal security, the user ID you give at installation if using advanced levels of security.

6.2 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 `<installation>/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 (not root) can and should run all diagnostic activities. This insures that any files or resources created are owned by the owner of the installation, 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.

6.2.1 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.

If you have any trouble starting Caché, view the console log from the **[Home] > [System Logs] > [View Console Log]** page of the System Management Portal 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 <installation>/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.

6.2.2 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 `<installation>/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.

6.2.3 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/tty/06
Terminal =>
Message =>
```

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

6.3 UNIX File Permissions

The permissions for files and directories are:

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
cforges         -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 dataset 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 dataset 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
```


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, SYSLCK, 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. Caché does not install a private Web server on a OpenVMS instance; therefore 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.

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 section describes the procedure for setting up the CSP Gateway and Web server environment on a Windows host and the next section provides additional configuration steps for more complex configurations:

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

7.3.1 Configuring the Portal on Windows

Though you can also use the portal from a UNIX-based Web server, these instructions describe the process for accessing the Caché instance on an OpenVMS system from a remote Web server on Windows:

1. Verify you have a supported Web server installed and running on the Windows machine.
2. Install a minimal Caché shell on the Web server machine. The detailed installation procedure is described in the “[Installing Caché on Windows](#)” chapter of the *Caché Installation Guide*. During the installation process, follow these instructions:
 - 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.

3. Once you have installed Caché on a Windows client you must now configure the CSP Gateway to access the OpenVMS-based Caché server. Navigate to the CSP Gateway Configuration page by pointing a browser to:

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

4. Click **Server Access** and **Add Server** to create a configuration to connect to the remote Caché server. For this server configuration, fill in the **Server Name**, **TCP/IP Address**, and **TCP/IP Port** fields with the name, IP address, and SuperServer port of the OpenVMS Caché server. Click **Save Configuration**.
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 of Caché.

7.3.2 Advanced Configuration

To properly use other CSP applications and Caché features, configure the following additional settings from the **[Home] > [Configuration] > [Advanced Settings]** page of the portal. You can enter `web` in the **Filter** box to shorten the settings list:

- **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** remote setting).
- **WebServerPort** — The port number of the Web server (equivalent to the **Web Server Port** remote setting).
- **WebServerURLPrefix** — The name of the Caché server instance (equivalent to the **CSP Server Instance** remote setting).

See the “[Connecting to Remote Servers](#)” chapter of this guide for more details in accessing an OpenVMS instance remotely.

For additional information, see the following:

- [Using Caché Server Pages with a Remote Web Server](#) section of the “Connecting to Remote Servers” chapter of this guide.
- “[Web Servers for Microsoft Windows](#)” chapter of *Installing and Configuring the CSP Gateway*

- For managing a clustered system or more complicated configurations, see the [Accessing CSP on Multiple Caché Servers](#) section of the “Connecting to Remote Servers” chapter of this guide or contact the [InterSystems Worldwide Response Center](#) (WRC).

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: Users of the Mac OS X platform can also use **SSH** from a command prompt on Windows to connect to their OS X machine.

You can also use Caché Server Pages (CSP) with a remote Web server. This is a requirement 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.

See the following sections for details:

- [Define a Remote Server Connection](#)
- [Using Caché Server Pages with a Remote Web Server](#)
- [Accessing CSP on Multiple Caché Servers](#)
- [Additional Resources](#)

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. 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.

Input Field	Description
Port	The port number of the SuperServer. The default port number is 1972.
Telnet Port	The port number of the telnet connection. The default port number is 23.
Web Server Port	The port number of the Web server. The default port number is 8972.
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.
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 Password or Kerberos .*
Connection Security Level	For <i>Kerberos</i> * only. Choose either Kerberos , Kerberos with Packet Integrity , or Kerberos with Encryption .
Windows Caché Telnet Server	For <i>Kerberos</i> * only. Select this check box if you are defining a connection to a Windows server.
Service Principal Name	For <i>Kerberos</i> * only. This field is pre-filled with the recommended service principal name format “cache/<FQHN>” (<i>FQHN</i> is the fully qualified host name) for the IP address you enter.

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

Important: The Caché SuperServer must be running on the remote machine and its port must be open on your firewall to use the Cube utilities on that system.

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

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](#).

8.1.2 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] > [Advanced Settings]** page of the portal. Enter **web** in the **Filter** box to shorten the settings list:

- **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).

8.2 Using Caché Server Pages with a Remote Web Server

This section discusses how to set up a Web server and the CSP Gateway to provide access to a CSP application installed on a remote Caché server. The instructions refer to the Web server as *Machine W* and to the computer running Caché as *Machine C*.

The examples are for a Windows installation and an IIS Web server, but you can also use these steps for a UNIX-based Web server host and an Apache Web server. The setup includes the following procedures:

1. [Install Caché Shell on Web Server Machine](#)
2. [Configure the CSP Gateway](#)
3. [Create Directories on Web Server](#)
4. [Add Virtual Directories to IIS](#)

8.2.1 Install Caché Shell on Web Server Machine

Install a minimal Caché shell on the Web server machine, *Machine W* where IIS is running. Reference the [Caché Installation Guide](#) if you need more detailed information. During the installation process, follow these instructions:

- 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 click **Next**.

This creates the CSP directory structure on *Machine W* and creates virtual directory references for the /CSP and /CSP/Bin files in IIS.

8.2.2 Configure the CSP Gateway

Next, adjust the CSP Gateway Configuration on *Machine W*. Though the CSP Gateway configuration information is stored in the csp.ini file, always use the Caché Server Pages Web Gateway Management application to update the configuration:

1. Navigate to the CSP **Web Gateway Management** home page by pointing a browser to <http://localhost/csp/bin/Systems/Module.czw> (bookmarking this URI is helpful).
2. Click **Server Access** in the left-hand menu. The Caché installation pre-configures a LOCAL server to connect to the Caché instance on the local machine, *Machine W*.

The name of the Caché server in the Gateway configuration does not need to match the name of the Caché instance to which it connects. The LOCAL configuration connects to the locally installed instance of Caché regardless of what that instance is actually named.

3. Create a new server name to represent the Caché instance running on the remote machine, *Machine C*:
 - a. Click **Add Server**.
 - b. Enter the **TCP/IP Address** and **TCP/IP Port** of the remote Caché server on *Machine C*.
 - c. Modify the **Connection Security** settings to match the level of authentication expected by *Machine C* for CSP Gateway connections.
 - d. Click **Save Configuration**.

4. Click **Application Access** in the left-hand menu to associate the path to the CSP application on the remote Caché server, *Machine C*, with the server configuration previously created for *Machine C*. The default paths are predefined for / and /csp.
5. Create a new application path to represent the CSP application running on the remote machine, *Machine C*. You can either copy an existing configuration (such as /csp) or click **Add Application** to manually create a new path configuration. The path you create for the application must match that defined for the application in the Caché configuration on *Machine C*.

For example, the default path to the System Management Portal is /csp/sys. If you are creating a new application choose your own path name. For example: /myapp or /csp/myapp. Having created the new path, modify the **Default Server** parameter for the path such that it takes the value of the Caché server configuration that you previously set up for *Machine C*. Finally, save the new path configuration.

The application paths in the previous steps correspond to requests for CSP pages in the equivalent locations. For example:

```
http://domain.com/myapp/login.csp
http://domain.com/csp/myapp/login.csp
http://domain.com/csp/sys/login.csp
```

Inheritance is applied in a hierarchical fashion. Consider the following request:

```
http://domain.com/csp/newapp/login.csp
```

The application path configuration for /csp/newapp is used if it exists. If not, the configuration defined for /csp is used instead.

The CSP Gateway installation procedures configure the hosting Web server to recognize the /csp virtual path. Typically, these same settings also apply to directories placed under /csp, (for example /csp/myapp). If you create a new path (such as /myapp) you must configure the Web server to recognize this new virtual path. See the following [Create Directories on Web Server](#) and [Add Virtual Directories to IIS](#) sections for details.

8.2.3 Create Directories on Web Server

If your CSP pages refer to static components held in the local path, create directories on *Machine W* to represent your application path. These directories exist solely to hold static content such as image files. You *do not* have to place any CSP files here; they reside on *Machine C*.

Under the directory CacheSys\CSP on *Machine W* (CacheSys is the default installation directory name), create \Samples and \User directories. Also create directories to represent other paths which may contain static components referenced in CSP pages. The example in the previous section, requires you to create a directory for \myapp.

8.2.4 Add Virtual Directories to IIS

The instructions in this section apply specifically to the IIS (Internet Information Services) Web server.

The installation procedure for the CSP Gateway configures the virtual directory /csp for CSP applications. If all of your applications are under this virtual directory (for example, /csp/myapp), you do not need to add virtual directories.

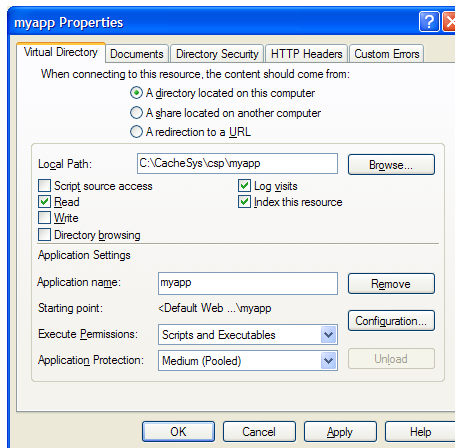
If you do not have an application path resembling /myapp in the previous examples, ignore this step. You must set it up with properties similar to the /csp virtual directory which is automatically created for you during the CSP Gateway installation.

1. Navigate to the **Internet Information Services** management dialog box, which is likely accessible from the **Administrative Tools** menu of the Windows Control Panel.
2. Expand the control trees and right-click **Default Web Site**.
3. Point to **New** and click **Virtual Directory** to create a new directory record with the following values:

```
Alias:                myapp
Directory:           c:\CacheSys\csp\myapp
Access Permissions:  Select the Execute check box
```

4. Either click **Save** and **Apply** all changes, or if you are using the wizard, click **Finish**.

Once you create the virtual directory record, right-click its name under **Default Web Site** and click **Properties**. The **Virtual Directory** tab in the **Properties** dialog box, looks similar to the following figure:



Restart IIS to apply these changes.

8.3 Accessing CSP on Multiple Caché Servers

The need to configure a single Web server to access the same CSP application on a number of different Caché servers often arises with the Systems Management Portal. This is particularly true of OpenVMS-based Caché installations in which clustered environments contain many instances of Caché.

This section uses the System Management Portal as the example application; however, you can apply the same procedures to any CSP application that is distributed over multiple Caché instances.

The portal application is usually called via:

```
http://domain.com/csp/sys/UtilHome.csp
```

One method for accessing this same application on multiple servers is to factor a name representing the individual Caché servers into the application path. For example:

```
http://domain.com/cache1/csp/sys/UtilHome.csp
http://domain.com/cache2/csp/sys/UtilHome.csp
```

Where cache1 and cache2 effectively represent separate instances of Caché.

You can configure this using the CSP Gateway Management application as follows:

1. Access the configuration application as shown below

```
http://localhost/csp/bin/Systems/Module.cxw
```

2. Click **Server Access** to create a server configuration for cache1 and cache2.
3. Click **Application Access** to create the application paths /cache1/csp/sys/ and /cache2/csp/sys/.

Set the **Default Server** for path /cache1/csp/sys/ to be the name of the server configuration set up for cache1 in the previous step.

Set the **Default Server** for path /cache2/csp/sys/ to be the name of the server configuration set up for cache2 in the previous step.

4. Within the configuration for the two remote Caché servers, cache1 and cache2, ensure that the paths to the portal are defined as paths /cache1/csp/sys/ and /cache2/csp/sys/ respectively.
5. If the Web server is IIS then set up virtual directories for /cache1 and /cache2 as described in the [Add Virtual Directories to IIS](#) section.

An alternative method for accessing an application on multiple servers is to use virtual host arrangements. Virtual hosts are a common feature in Apache Web server configurations and are straightforward to set up in this server environment. For example, consider two virtual hosts, each listening on a separate TCP port:

```
http://virtual_host1:81/csp/sys/UtilHome.csp  
http://virtual_host2:82/csp/sys/UtilHome.csp
```

Both virtual_host1 and virtual_host2 are served by the same Web server and CSP Gateway.

The following shows the Apache configuration (httpd.conf) for this arrangement:

```
<VirtualHost virtual_host1:81>  
    ServerName virtual_host1  
</VirtualHost>  
  
<VirtualHost virtual_host1:82>  
    ServerName virtual_host2  
</VirtualHost>
```

Configure the use of these virtual hosts using the CSP Gateway Management application as follows:

1. Navigate to the Caché Server Pages Gateway Management home page by pointing a browser to:

```
http://localhost/csp/bin/Systems/Module.cxw
```

2. Click **Server Access** to create a server configuration for cache1 and cache2.

3. Click **Application Access** to create the application paths `//virtual_host1/csp/sys/` and `//virtual_host2/csp/sys/`.

Note the use of the double forward-slash (`//`) to introduce the virtual host name.

Set the **Default Server** for path `//virtual_host1/csp/sys/` to be the name of the server configuration set up for `cache1` in the previous step.

Set the **Default Server** for path `//virtual_host2/csp/sys/` to be the name of the server configuration set up for `cache2` in the previous step.

4. No changes are required in the configuration of the two remote Caché servers. The application path for the portal remains as `/csp/sys/` in both cases.

8.4 Additional Resources

For additional information, see the following:

- [Install Caché on a Web Server](#) section of the “Installing Caché on OpenVMS” chapter of the *Caché Installation Guide*.
- [Installing and Configuring the CSP Gateway](#)
- For managing a clustered system or more complicated configurations, contact the [Inter-Systems Worldwide Response Center](#) (WRC).

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. 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	ccontrol start <instname>
Connecting to a Caché instance	csession <instname>
Stopping a Caché instance	ccontrol stop <instname>

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
csession <instname> -"B"	Provides system administrator login to perform system tasks such as setting up licensing.
csession <instname> -"U" "namespace"	Specifies the login namespace.
csession <instname> "[label[+offset]]^routine"	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 multiple instances of Caché using the **ccontrol** command. The **ccontrol** command supports a number of functions and has the following syntax:

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

Caché ccontrol Command and Functions

Command	Description
ccontrol start <instname> [args]	Brings up an instance.
ccontrol stop <instname> [args]	Shuts down an instance.
ccontrol force <instname>	Forces an instance down.
ccontrol list	Lists all installed instances.
ccontrol all	Shows compact summary of all instances.
ccontrol stat <instname> [args]	Retrieves system statistics.
ccontrol help [start stop force]	Gives function-specific help for the start, stop, and force functions.
ccontrol load <instname>*	Runs the procedure for loading images.
ccontrol unload <instname>*	Runs the procedure for unloading images.
ccontrol rename <instname> {newname}	Renames the configuration file.

* Functions for OpenVMS only.

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

The optional arguments (indicated by [args]) 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)

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. Invoke the **ccontrol help** command to obtain the most recent description of the interface with these scripts.

On Windows to display the most current help file for running the **ccontrol** command, run **ccontrol help** from the Caché instance bin directory For example:

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

The command displays the CcontrolHelp.html file in your browser; the file is in the <cache-install-dir>\Help directory (CacheSys\Help by default).

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

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

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 dis-

tributed data protocols that are supported for interoperation 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
- [Web server port number](#) — 8972 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] > [Advanced Settings]** page 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 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 (yyyymmdd.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.

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`. On UNIX systems, you may specify a UNIX raw partition in your system configuration.

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 High Availability 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 High Availability 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 Multidimensional Storage*.

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
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 High Availability 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 High Availability 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](#).