



# Caché Additional Configuration Settings Reference

Version 2018.1  
2019-09-20

*Caché Additional Configuration Settings Reference*

Caché Version 2018.1 2019-09-20

Copyright © 2019 InterSystems Corporation

All rights reserved.



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



InterSystems IRIS Data Platform, InterSystems IRIS, InterSystems iKnow, Zen, and Caché Server Pages are trademarks of InterSystems Corporation.

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

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

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

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

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

For Support questions about any InterSystems products, contact:

**InterSystems Worldwide Response Center (WRC)**

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: [support@InterSystems.com](mailto:support@InterSystems.com)

# Table of Contents

<b>About this Book</b> .....	<b>1</b>
<b>1 Caché Additional Configuration Settings</b> .....	<b>3</b>
1.1 Advanced Memory Settings .....	3
1.2 Compatibility Settings .....	8
1.3 Device Settings .....	14
1.3.1 Devices .....	14
1.3.2 Magnetic Tape Devices .....	15
1.3.3 Device SubTypes .....	15
1.3.4 IO Settings .....	16
1.3.5 Telnet Settings .....	17
1.4 Monitor Settings .....	17
1.5 National Language Settings .....	18
1.5.1 Locale Definitions .....	18
1.5.2 Import Locales or Tables .....	20
1.6 Source Control Settings .....	20
1.7 SQL and Object Settings .....	20
1.7.1 General SQL Settings .....	20
1.7.2 TSQL Compatibility Settings .....	26
1.7.3 ISQL Compatibility Settings .....	26
1.7.4 FileMan Conversion Settings .....	27
1.7.5 System-Defined DDL Mappings .....	29
1.7.6 User-Defined DDL Mappings .....	29
1.8 Startup Settings .....	30
1.9 Task Manager Email Settings .....	34
1.10 Zen Reports .....	35
1.10.1 Zen Report Render Servers .....	35
1.10.2 Zen Report Settings .....	37



# About this Book

This book describes settings available on the **Additional Settings**, **Device Settings**, and **SQL and Object Settings** pages in the **Configuration** section of the Management Portal (**System Administration > Configuration**). It contains the following sections.

- [Compatibility Settings](#)
- [Device Settings](#)
  - [Devices](#)
  - [Magnetic Tape Devices](#)
  - [Device Subtypes](#)
  - [IO](#)
  - [Telnet](#)
- [Advanced Memory Settings](#)
- [Monitor Settings](#)
- [National Language Support Settings](#)
- [Source Control Settings](#)
- [SQL Settings](#)
  - [SQL General Settings](#)
  - [TSQL Compatibility Settings](#)
  - [FileMan Conversion Settings](#)
  - [System-Defined DDL Mappings](#)
  - [User-Defined DDL Mappings](#)
- [Startup Settings](#)
- [Task Manager E-mail Settings](#)
- [Zen Report Settings](#)



# 1

## Caché Additional Configuration Settings

This chapter describes some configuration settings that are available from the **System Administration > Configuration** page in the Management Portal.

Under the **System Administration > Configuration** page are the options:

- [SQL and Object Settings](#)
- [Device Settings](#)
- [National Language Support Settings](#)
- [Zen Reports](#)

Under the **System Administration > Configuration > Additional Settings** page are the options:

- [Compatibility Settings](#)
- [Advanced Memory Settings](#)
- [Monitor Settings](#)
- [Source Control Settings](#)
- [Startup Settings](#)
- [Task Manager Email Settings](#)

### 1.1 Advanced Memory Settings

The **Advanced Memory** page (**System Administration > Configuration > Additional Settings > Advanced Memory**) contains a number of memory-related and other settings, as listed in the following.

**Important:** The **Memory and Startup** page (**System Administration > Configuration > System Configuration > Memory and Startup**) lets you allocate memory to routine and databases caches; when Caché is first installed, this cache allocation is set to **Automatically**. *This setting is not appropriate for production use.* Before deploying the system for production use or performing any tests or benchmarking intended to simulate production use, you must manually create an appropriate memory allocation by selecting **Manually**. For details, see [Memory and Startup Settings](#) in the “Configuring Caché” chapter of the *Caché System Administration Guide*.

### BackoffDisabled

On failure to allocate memory, do not retry with reduced amount. If the memory cannot be allocated as its configured size, startup is aborted. For more information see the *memlock* entry in the “[List of Sections and Parameters section](#)” of the “Caché Parameter File Reference”.

### ConsoleFile

File in which to log system console messages. If no value is specified for ConsoleFile, Caché writes to the file *cconsole.log* in the Caché system management directory. See [MaxConsoleLogSize](#).

The ConsoleFile and [MaxConsoleLogSize](#) settings apply on all platforms.

### LargePagesDisabled

On platforms supporting large pages or huge pages, disable use of them for shared memory. For more information, see the *memlock* entry in the “[List of Sections and Parameters section](#)” of the “Caché Parameter File Reference”.

### LargePagesRequired

On platforms supporting large or huge pages (Windows, AIX, and Linux), require use of them for shared memory. For more information, see the *memlock* entry in the “[List of Sections and Parameters section](#)” of the “Caché Parameter File Reference”.

### LibPath

UNIX® systems only. Sets the `LD_LIBRARY_PATH` environment variable to paths to search for third-party shared libraries. If you modify this setting, you must restart the system to make it active.

On macOS, if you have enabled System Integrity Protection (SIP), it may ignore the `DYLD_LIBRARY_PATH` variable executing programs in the system directories.

### LineRecallBuffer

Total size (in bytes) of all input strings to be stored in the command line/read line buffer. The default is 1024 bytes. The range is 0–8192 bytes. See [LineRecallEntries](#).

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

### LineRecallEntries

Maximum number of entries held in the command line/read line recall buffer, subject to the space limitation in the [LineRecallBuffer](#) setting. The default is 32 entries. The range is 0–256 entries.

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

### LockSharedMemory

True or false. When true, shared memory is locked in memory to prevent paging. The default is false. If you edit this setting, you must restart Caché to apply the change. For more information, see the *memlock* entry in the “[List of Sections and Parameters section](#)” of the “Caché Parameter File Reference”.

### LockTextSegment

True or false. When true, the text segment (the Caché executable code space) is locked into shared memory. The default is false. If you edit this setting, you must restart Caché to apply the change. For more information, see the *memlock* entry in the “[List of Sections and Parameters section](#)” of the “Caché Parameter File Reference”.

## MaxServerConn

Maximum number of ECP clients that can access this system simultaneously. This is the maximum number of connections that this system may accept when acting as an ECP server. The range is 0–254 clients. The default is 1 client.

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

## MaxServers

Maximum number of ECP servers that can be accessed from this system. This is the maximum number of connections that this system can establish when acting as an ECP client. The range is 0–254 servers. The default is 2 servers.

## Path

(UNIX only.) The default `Path` used for all processes started during cache startup is `Path=/usr/bin:/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin`. You can append more directories to this path using the `Path` variable.

All the processes started during Phase 2 of Startup (which runs the `STU1.mac` routine) search in the base `Path` followed by any directories that you add to the `Path` here.

## VMSConsoleTerminal

This parameter is no longer supported.

## ZFSize

The \$ZF heap is used for output parameters in a callout function using \$ZF. For details, see the chapter “[Creating a Caché Callout Library](#)” in *Using the Caché Callout Gateway*.

Two parameters configure the \$ZF heap: `ZFString` and `ZFSize`.

- `ZFSize` is the number of bytes Caché allocates for the \$ZF heap for all purposes. The \$ZF heap consists of the total number of bytes allocated in virtual memory for all \$ZF input and output parameters, including the space for strings allowed by the first value. The range is 0–270336 bytes. The default is 0. A value of 0 tells Caché to determine an appropriate value for `ZFSize` automatically, based on the value of `ZFString`.

The formula for calculating `ZFSize` based on `ZFString` is as follows:

$$\text{ZFSize} = (\text{BytesPerCharacter} * \text{ZFString}) + 2050$$

For example, suppose `ZFString` has the default value, 32767:

- If you are using Unicode, a single character is 2 bytes. An appropriate value for `ZFSize` is then 67584 (or  $2 * 32767 + 2050$ ) bytes.
- On UNIX®, a single character is 4 bytes. An appropriate value for `ZFSize` is then 133118 (or  $4 * 32767 + 2050$ ) bytes.

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

## ZFString

The \$ZF heap is used for output parameters in a callout function using \$ZF. For details, see the chapter “[Creating a Caché Callout Library](#)” in *Using the Caché Callout Gateway*.

Two parameters configure the \$ZF heap: `ZFString` and `ZFSize`.

- [ZFString](#) is the number of characters Caché allows for a single string parameter on the \$ZF heap. How many bytes this actually requires depends on whether you are using 8-bit characters, Unicode (2-byte characters), or 4-byte characters on UNIX®. The range is 0–32767 characters. The default is 0. A value of 0 tells Caché to determine an appropriate value for [ZFString](#) automatically. The default in this case is 32767 characters.

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

### errlog

Maximum number of entries in the Caché system error log (see [Caché System Error Log](#) in the “Monitoring Caché Using the Management Portal” chapter of the *Caché Monitoring Guide* for more information). The log file expires old entries as this limit is reached. The default is 500 entries. The range is 10–10000 entries.

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

### gmheap

Size (in kilobytes) of the generic memory heap (also known as the shared memory heap or SMH) for Caché.

The `gmheap` setting configures total shared memory available for use by the subsystems of a Caché instance. Shared memory is allocated from this total as needed for particular purposes (such as global mapping, database name and directory information, the security system, and so on). The shared memory in use by a given subsystem at a given time may be less than what is currently allocated.

Shared memory allocation is shown on the **Shared Memory Heap Usage** page (navigate to the **System Operation > System Usage** page and click the **Shared Memory Heap Usage** link); see [Generic \(Shared\) Memory Heap Usage](#) in the “Monitoring Caché Using the Management Portal” chapter of the *Caché Monitoring Guide* for more information. Although this page displays memory allocation and use in bytes, bear in mind that shared memory is allocated in pages.

The default size of `gmheap` is 37568 kb (53952 kb for IBM AIX® and HP-UX). (Note that in some cases, the actual maximum amount of shared memory available for allocation may be more than what is specified by `gmheap`. For example, more than the amount specified may be allocated to compensate for the needs of multiple CPUs.)

Under some circumstances it may be necessary to increase `gmheap` to make enough shared memory available, for example in the following situations:

- Adding language models to iKnow

The default `gmheap` value is enough for iKnow language models `en` and `es`. If you are installing other iKnow language models, increase `gmheap` by the amounts shown in the table below. For each Asian locale, add one extra megabyte (1024 kilobytes) to the total `gmheap` value.

Language	Increase gmheap by
de	4.5MB
en	3.3MB
es	12MB
fr	4.4MB
nl	3.6MB

- Restoring journal files

To ensure optimal performance during a journal restore, InterSystems recommends that you increase the generic memory heap size; see [Restore Journal Files](#) in the “Journaling” chapter of the *Cache Data Integrity Guide* for more information.

- When parallel SQL query execution is in use

Parallel query execution uses additional shared memory from the generic memory heap, and an increase in `gmheap` may therefore be required to optimize parallel query performance. See [Shared Memory Considerations](#) in the “Optimizing Query Performance” chapter of the *Caché SQL Optimization Guide* for more information.

The `locksiz` setting configures the portion of total available shared memory that can be specifically allocated for managing locks (the lock table). `locksiz` is a subset of `gmheap`, and the remainder of `gmheap` is what is available for all other subsystems, so it is important that `gmheap` and `locksiz` be sized in consideration of this relationship, and that when `locksiz` is increased, `gmheap` is also increased proportionally.

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

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

### **ijcbuf**

Number of bytes allocated for each InterJob Communication (IJC) buffer. The default is 512 bytes. The range is 512–8192 bytes. For details, see the “[Interprocess Communication](#)” chapter in the *Caché I/O Device Guide*. Also see [ijcnum](#).

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

### **ijcnum**

Number of InterJob Communication (IJC) devices. Each device corresponds to one InterJob Communication buffer of the size defined by `ijcbuf`. The default is 16 devices. The range is 0–256 devices. For details, see the *Caché I/O Device Guide* chapter [Interprocess Communication](#).

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

### **jrnbufs**

Amount of memory allocated for journal buffers in MB, from 8 to 1024. The default is 64.

### **locksiz**

Size (in bytes) of shared memory allocated for locks. The system rounds up the value to the next multiple of 64 kilobytes. The default is 16777216 bytes. (On the IBM AIX and HP-UX platforms, the default is 33554432 bytes.) This setting is a subset of the shared memory allocated for all purposes by the `gmheap` setting; the range is 65536 bytes up to the `gmheap` setting, but as a practical matter `locksiz` should be only a fraction of `gmheap` so that shared memory is available for other purposes. If you need more room for the lock table, increase `gmheap` and then increase `locksiz`.

If you edit this setting, changes take place immediately.

### **netjob**

Allow remote job requests to run on this server.

- True - Run incoming remote job requests on this server. (default)
- False - Reject incoming remote job requests.

If you edit this setting, changes take place immediately.

**nIstab**

Enter the maximum number of collation tables, from 0–64. The default is 30. This instructs Caché to reserve space for that many tables at startup.

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

**targwijsz**

Set the desired size of the WIJ file. For more information, see the [Write Image Journaling and Recovery](#) chapter of *Caché Data Integrity Guide*.

**udevtabsiz**

Maximum size (in bytes) of the device table. This is the table that maps device numbers (traditional logical unit numbers) to device names, so that ObjectScript code can open devices by number. The default is 24576 bytes. The range is 0–65535 bytes.

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

**vectors**

Deprecated. No longer used by Caché.

## 1.2 Compatibility Settings

The **Compatibility** page (**System Administration > Configuration > Additional Settings > Compatibility**) contains the following settings:

**ASyncDisconnectErr**

This setting modifies the behavior of Caché when [DisconnectErr](#) is enabled.

- True (selected)- The process receives a <DSCON> error at the next read or write command.
- False (default) (cleared) - The process receives an asynchronous <DSCON> error at the time a disconnect occurs on the device. This error occurs at the next command executed. Hang commands are interrupted.

`ASyncDisconnectError` is only applicable to Telnet connections on Windows. It has no effect on any other device type or operating system. If [DisconnectErr](#) is false, then `ASyncDisconnectError` has no effect.

**AsynchError**

Enable (selected) or disable (cleared) processes to receive asynchronous errors.

- True (selected) (default) - Caché processes can receive asynchronous errors.
- False (cleared) - Caché processes cannot receive asynchronous errors.

**BreakMode**

True (selected) or false (cleared). Programmer mode response to the BREAK command.

Caché programs can execute in two modes, depending on how Caché is entered: [application mode](#) and [programmer mode](#).

The `BreakMode` setting controls how a Caché process in programmer mode responds when it encounters a [BREAK](#) command that has no argument.

- True (selected) (default) - Caché enters the debugger or returns to the direct mode prompt with a <BREAK> error.
- False (cleared)- The BREAK command is ignored.

Application mode jobs always ignore BREAK commands with no arguments.

### CollectResourceStats

Controls whether system resource statistics are updated or not (seize, nseize, aseize, bseize). The default is that they are not collected.

- When 1 (true), Caché collects system resource statistics.
- When 0 (false), Caché does not collect system resource statistics. The default is false.

For more information on system resource statistics, see the appendixes “[Monitoring Caché Using the cstat Utility](#)” and [Monitoring Caché Using Web Services \(Enumresource\)](#) in the *Caché Monitoring Guide*.

### DisconnectErr

How Caché responds to a disconnect of the principal I/O device.

- True (selected) - the process receives a <DSCON> error when a disconnect is detected during a Caché Write or Read command.
- False (cleared) - the process exits without reporting an error to the application when a disconnect is detected. (default)

Note that, if ErrorOnDisconnect is enabled, a process continues to execute after its principal device has been disconnected. It is the responsibility of the application to detect the disconnect condition and exit gracefully.

Use care when enabling ErrorOnDisconnect. The application must be prepared to recognize the <DSCON> error and handle it appropriately in error traps.

ErrorOnDisconnect is only applicable to TCP devices and to terminal devices where a disconnect can be recognized. Examples are modem controlled terminals and Windows Telnet, and Windows local cterm (TRM:) connections. ErrorOnDisconnect is only applicable to the principal device.

### FileMode

Create a file if it does not exist when called with Write or Read/Write.

- True (selected) - If a file is opened for writing that does not exist, a new file is created.
- False (cleared) - If a file is opened for writing that does not exist, a new file is not created unless the N parameter was provided with the OPEN command.

Suppose Caché encounters an OPEN command such as:

```
OPEN "file.x": "WS"
```

When FileMode is true, the new file is created automatically even though the N parameter is not specified with the OPEN command. The result when FileMode is true is equivalent to adding the N parameter to each OPEN command, so that:

```
OPEN "file.x": "WNS"
```

is equivalent to:

```
OPEN "file.x": "WNS"
```

On the other hand, when Caché encounters an OPEN command *and* no N parameter is provided *and* the file does not already exist, then if FileMode is false there is no result from the OPEN command except that the process hangs until interrupted.

### GlobalKillEnabled

Deprecated. Enable (selected) or disable (cleared) KILL of an unsubscripted global.

- True (selected) - a KILL of an unsubscripted global is allowed, so you can kill all subscripts of a global with a single kill instead of killing them individually. (default)
- False (cleared) - the KILL of an unsubscripted global results in a <PROTECT> error.

### IEEEError

Enables or disables \$DOUBLE returning INF and NAN values system-wide. This property sets the \$DOUBLE function return-value behavior system-wide.

The property controls the issuing of INF, -INF, and NAN when a \$DOUBLE numeric operation cannot be resolved to a numeric value. It does not control the issuing of INF, -INF, and NAN in all cases. \$DOUBLE always returns INF, -INF, or NAN when you supply one of these strings as the input value, regardless of this property. Mathematical operations on \$DOUBLE numbers that result in an INF, -INF, or NAN are controlled by this property. These include arithmetic operations, exponentiation, and logarithmic and trigonometric functions.

- When 1 (true), \$DOUBLE generates Caché errors for unresolvable IEEE floating point conversions. The default is true.
- When 0 (false), \$DOUBLE returns INF (infinity), -INF, and NAN (Not A Number) for unresolvable IEEE floating point conversions.

### LineRecall

Enable (selected) or disable (cleared) command line recall for READ commands.

The selection determines whether the line recall feature is active only for command prompts, or for both command prompts and READ commands.

- True (selected) - READ commands *and* command prompts can use the line recall feature. (default)
- False (cleared) - only command prompts can use line recall.

### LogRollback

Enable (selected) or disable (cleared) logging for transaction rollbacks.

- True (selected) - Caché logs transaction rollbacks to the console log file (that is, cconsole.log in the Caché system management directory, or the alternate *filename.log* named by the [ConsoleFile](#) setting).
- False (cleared) - it does not log transaction rollbacks. (default)

### MVDefined

Sets MVBasic handling of undefined variables system-wide. This setting defines MVBasic behavior when it encounters a reference to an undefined variable.

- When 1 (true), if an MVBasic routine references an undefined variable, the system substitutes an empty string for the variable without showing an error.
- When 0 (false), if an MVBasic routine references an undefined variable, the system generates an error. The default is false.

## NodeNameInPid

Defines the behavior when Caché makes a reference to the special variable \$JOB.

- True (selected) - \$JOB returns the process ID number of the current process, concatenated to the nodename.
- False (cleared) - \$JOB returns only the process ID number. (default)

## NullSubscripts

Allow (selected) null subscripts on global references

- True (selected) - null subscripts are allowed on global references.
- False (cleared) - a null subscript causes a <SUBSCRIPT> error. (default)

## OldZU5

- True (selected) - Switching to the same namespace via [\\$ZU\(5\)](#) or [ZN](#) clears the globals vector cache.
- False (cleared) (default) - Switching to the same namespace is a NOOP.

## OpenMode

0 or 1 specifies the read/write mode to use when opening sequential files and no mode is specified in the OPEN command.

- 0 for Read.
- 1 for Read-Write.

## PopError

Defines when to pop error handlers off the stack.

True (selected) - Pop the \$ZTRAP error handler off the stack when an error is triggered (DSM compatibility mode)

False - (cleared) Normal behavior: A \$ZTRAP error handler stays active when the error handler is invoked. (default)

When a \$ZTRAP error handler is invoked by the system, that error handler remains on the stack of established error handlers. Thus, if an error occurs when the error handler is executing, that error handler attempts to invoke itself, receives the same error again, and enters an infinite loop, unless that error handler explicitly sets \$ZTRAP to a new value. When a \$ZTRAP error handler is invoked in DSM, the error handler is removed from the stack. Thus, if an error occurs while the error handler is executing, that error is handled by the previous error handler on the stack.

## RefInKind

Determines the result of \$NAME and \$QUERY when an extended global reference is given as the argument.

- True (selected) - \$NAME and \$QUERY return the global without reference to where it is on the network.
- False (cleared) - the result is also an extended reference. (default)

If you change the value of this parameter, the change applies to processes started after the change, but not for processes that were already running when you made the change.

## ScientificNotation

This setting enables or disables using the lowercase "e" as scientific notation symbol system-wide.

- When 1 (true), Caché uses the lowercase "e" as scientific notation symbol. The default is true.

- When 0 (false), Caché does not use the lowercase "e" as scientific notation symbol.

### SetZEOF

Determines the behavior when Caché encounters an unexpected end-of-file when reading a sequential file.

- True (selected) - Caché sets the special variable \$ZEOF to indicate that you have reached the end of the file.
- False (cleared) - Caché throws an <ENDOFFILE> error instead. (default)

### StopID

This parameter is no longer supported.

### ShutDownLogErrors

This setting controls Caché behavior during execution of its SHUTDOWN procedure.

- True (selected) — During shutdown Caché logs error information from ^SYSLOG into the console log file (that is, cconsole.log in the Caché system management directory, or the alternate *filename.log* named by the [ConsoleFile](#) setting).
- False (cleared) — Caché does not log error information from ^SYSLOG into the console log file. (default)

### SwitchOSdir

Specifies what happens to the current working directory (for accessing files by relative pathname, etc.) when you switch to a new namespace.

- True (selected) - if you change namespaces, the current working directory is changed to the directory of the default dataset for non-% globals of the new namespace. However, if this dataset is remote (networked to a different system), the current working directory is left unchanged. (default)
- False (cleared) - if you change namespaces, the current working directory remains unaltered no matter what namespace you switch to.

Suppose SwitchOSDirectory is set to false, *or* SwitchOSDirectory is set to true and the dataset is remote. In these cases, the current working directory does not change automatically as a result of changing the namespace, but you can always change the current working directory programmatically, using the [\\$ZUTIL\(168\)](#) function. [\\$ZUTIL\(168\)](#) works even when automatic switching during a namespace change has been inhibited. [\\$ZUTIL\(168\)](#) returns the name of the current working directory, and [\\$ZUTIL\(168,directory\)](#) sets the current working directory to the specified *directory* name regardless of the value of SwitchOSDirectory.

### SynchCommit

Every [TCOMMIT](#) command requests a flush of the journal data involved in that transaction to disk.

- True (selected) - TCOMMIT does not complete until the journal data write operation completes.
- False (cleared) - TCOMMIT does not wait for the write operation to complete. (default)

### TelnetNUL

Suppress/issue Telnet NUL at end-of-line during Telnet transmission. On output, a Telnet network virtual terminal (NVT) performs the following default end-of-line behavior: either issues a CR (carriage return character) followed by a LF (linefeed character), or issues a CR followed by a NUL character (if no LF is issued). The TelnetNUL setting affects the issuance of the NUL character in the second case. (This setting applies only to Windows; it has no effect on UNIX® and Linux, configurations, in which Telnet is supplied by the operating system vendor.)

- True (selected) - TelnetNUL suppresses (does not issue) a NUL character at end-of-line.
- False (cleared) - TelnetNUL issues a NUL character at end-of-line. Default.

### TruncateOverflow

Enable (cleared) or disable (selected) the <MAXNUMBER> error on numeric overflow.

Normally, when Caché encounters a number larger than 92372036854775807 E127 (or smaller than - 9223372036854775808 E127) it throws the <MAXNUMBER> error. You can prevent this by selecting the `TruncateOverflow` setting.

- True (selected) - the <MAXNUMBER> error is suppressed.
- False (cleared) - Caché throws the <MAXNUMBER> error as usual. (default)

### Undefined

Specifies how ObjectScript reacts when it attempts to fetch the value of a variable that has not been defined.

- 0 - Throw an <UNDEFINED> error. (default)
- 1 - If the undefined variable has subscripts, return a null string, but if the undefined variable is single-valued, throw an <UNDEFINED> error.
- 2 - Always return a null string.

### UseNagleAlgorithm

Enable (selected) or disable (cleared) the Nagle algorithm for Telnet.

The Nagle algorithm makes Telnet more efficient. It reduces the number of IP packets sent over the network by consolidating messages that are sent within a small time interval into a single IP packet. For details see <http://www.ietf.org/rfc/rfc896.txt>

- True (selected) — When the Nagle algorithm is enabled, the operating system waits some interval before actually committing the data from a send command, in the hopes that the application will call send again with more data that can be consolidated with the first.
- False (cleared) — Nagle algorithm is disabled.

### ViewPastData

Enable \$VIEW command to examine data outside of Caché memory area.

- True (selected) - \$VIEW command does not throw an error.
- False (cleared) - \$VIEW command throws an error (default)

### ZDateNull

Determines how a \$ZDATE call responds when triggered by an invalid value.

- True (selected) - \$ZDATE returns a null value.
- False (cleared) - \$ZDATE returns an error. (default)

### ZaMode

Determines how the ZALLOCATE (ZA) and ZDEALLOCATE (ZD) commands behave, according to Caché rules or DSM-11 rules:

- True (selected) - Uses DSM-11 rules, which means that ZA locks can only be unlocked by ZD and LOCK + locks can only be unlocked by LOCK.
- False (cleared) - Uses Caché rules, which means that ZA and ZD behave exactly like LOCK + and LOCK (default).

For details, see the section “[DSM-11 Language Compatibility](#)” in the “Open M Language Compatibility” chapter of *Using Caché ObjectScript*.

## 1.3 Device Settings

The **Device Settings** menu (**System Administration > Configuration > Device Settings**) contains the following settings:

- [Devices](#)
- [Magnetic Tape Devices](#)
- [Device SubTypes](#)
- [IO](#)
- [Telnet](#)

### 1.3.1 Devices

The **Devices** page (**System Administration > Configuration > Device Settings > Devices**) contains the following settings:

#### Devices

List of devices configured for this installation.

#### **Edit or Add Device Configuration Fields**

**Alias**-An alternate device ID for this device. All aliases must be unique. You can use this value as the *device* argument in an [OPEN](#) command.

**AlternateDevice**-The device ID of another device. The value entered for **AlternateDevice** must be a defined mnemonic such as the **Name** supplied for another device.

Specifying an **AlternateDevice** value for the device allows users of the %IS utility to specify “A” to tell Caché to use the alternate device. %IS is a general device selection utility for character-based applications. For details about %IS see the section “[Allowing Users to Specify a Device](#)” in the “I/O Devices and Commands” chapter of the *Caché I/O Device Guide*. The topic of most interest is “[%IS Mnemonics](#),” which describes the conventions for entering the “A” code for %IS.

**Description**-A text description of where the device is located. This field is for your own reference to help you identify what machine you are configuring.

**Name**-The mnemonic that is the defined device title or number.

**OpenParameter**-A colon-separated string that provides the *parameters*, *timeout*, and *mnespace* arguments for this device’s [OPEN](#) command. The syntax for the **OpenParameter** string is:

```
(parameters) : timeout : "mnespace"
```

Inside the parentheses for *parameters*, individual items are colon-separated, as follows:

```
param1 : param2 : param3
```

Resulting in:

```
(param1:param2:param3):timeout:"mnespace"
```

*timeout* and *mnespace* are optional, but if they are provided, the correct number of colons must separate them from previous entries in the **OpenParameter** string.

*parameters* must be contained within parentheses only if there is more than one parameter. If there are no parameters, or if there is only one parameter, the parentheses may be omitted from the string. Thus the following is a correct and complete **OpenParameter** string:

```
:timeout:"mnespace"
```

If provided, *mnespace* must be contained within double quotes, as shown.

For details about the **OPEN** command and its arguments, including a large variety of syntax examples, see the [Caché ObjectScript Reference](#).

**PhysicalDevice**-The physical name used to refer to the device. The **PhysicalDevice** value specifies the *device* argument for this device's **OPEN** command. The name can contain up to 128 alphanumeric characters; it can contain space characters as well. For example, for a printer you could enter the following, where *MYNAME* is the computer name.

```
|PRN|\\MYNAME\ISF-HP5SiMX7
```

Or:

```
|PRN|\\MYNAME\Canon PIXMA
```

**Prompt**-Choose an option: Show device prompt (the user sees the device selection prompt with the default device defined). Auto-use this device if it is the current device. Auto-use this device with predefined settings.

**SubType**-Used to refine the definition of your device subtypes. SubTypes specify terminal characteristics. They are used to create the appropriate **OPEN** command for the device. There should be subtype information for every terminal type.

**Type**-The type of device. Options: TRM=Terminal. SPL=Spooling device. MT=Magnetic Tape drive. BT=Cartridge tape drive. OTH=Any other device including printers and sequential files. The default depends on the device type.

The maximum length of most strings above is 128 characters. The exception is the **Description** string, which may be 256 characters long.

## 1.3.2 Magnetic Tape Devices

The **Magnetic Tape Devices** page (**System Administration > Configuration > Device Settings > Magnaetic Tape Devices**) contains the following settings:

### MagTapes

List of magnetic tape devices configured for this installation.

#### **Configuration Fields**

**Name**-The pathname to the device in the system. This string may be up to 128 characters long.

**System Device**-The name of the physical tape device.

## 1.3.3 Device SubTypes

The **Device SubTypes** page (**System Administration > Configuration > Device Settings > Device SubTypes**) contains the following settings:

## SubTypes

List of device subtypes configured for this installation.

### **Configuration Fields**

**Backspace**-The ASCII code that represents the backspace character on the selected device in the form  $\$C(\text{code1})$ . This setting is used by the Caché CHUI utilities.

**CursorControl**-The ASCII code that represents the cursor on the selected device in the form  $\$C(\text{code1})$ .

**EraseEOF**-The ASCII code that represents erasing the end of file character on the selected device in the form  $\$C(\text{code1},\text{code2}\dots)$ .

**EraseEOL**-The ASCII code that represents erasing the end of line characters on this device in the form  $\$C(\text{code1},\text{code2})$ .

**FormFeed**-The ASCII code that represents a form feed on the selected device in the form  $\#, \$C(\text{code1},\text{code2}\dots)$ . This setting is used by the Caché CHUI utilities.

**Name**-The name of the sub-type.

**RightMargin**-The number that represents the location of the right margin. Device output will wrap at that number of characters.

**ScreenLength**-The number of lines that comprise one screen or page for the device.

**ZU22Backspace**-The ASCII code that represents a backspace on the selected device in the form  $\$C(\text{code1})$ . This setting is used by Caché for Terminal output.

**ZU22FormFeed**-The ASCII code that represents a form feed on the selected device in the form  $\$C(\text{code1},\text{code2})$ . This setting is used by Caché for Terminal output.

All default values depend on the device type.

## 1.3.4 IO Settings

The **IO Settings** page (**System Administration > Configuration > Device Settings > IO Settings**) contains the following settings:

### **File**

Default for WRITE commands to a sequential file. When an OPEN or USE command includes no space argument, Caché uses the default for that device type.  $\^{\%}X364$  is the default space for a sequential file. You can use this default, or reset it as needed. See the “[Controlling Devices with Mnemonic Spaces](#)” chapter in the *Caché I/O Device Guide*.

If you edit this setting, the change applies automatically the next time you restart Caché.

### **MagTape**

Default for WRITE commands to magnetic tape. When an OPEN or USE command includes no space argument, Caché uses the default for that device type.  $\^{\%}XMAG$  is the default space for magnetic tape. You can use this default, or reset it by changing the value of as needed. See the “[Controlling Devices with Mnemonic Spaces](#)” chapter in the *Caché I/O Device Guide*.

If you edit this setting, the change applies automatically the next time you restart Caché.

## Other

Default for WRITE commands to device types other than magnetic tape, terminal, or sequential file. When an OPEN or USE command includes no space argument, Caché uses the default for that device type. ^%X364 is the default space for device types other than magnetic tape, terminal, or sequential file. You can use this default, or reset it as needed. See the “[Controlling Devices with Mnemonic Spaces](#)” chapter in the *Caché I/O Device Guide*.

If you edit this setting, the change applies automatically the next time you restart Caché.

## Terminal

Default for WRITE commands to a terminal device. When an OPEN or USE command includes no space argument, Caché uses the default for that device type. ^%X364 is the default space for a terminal. You can use this default, or reset it as needed. See the “[Controlling Devices with Mnemonic Spaces](#)” chapter in the *Caché I/O Device Guide*.

If you edit this setting, the change applies automatically the next time you restart Caché.

## 1.3.5 Telnet Settings

Caché Telnet settings apply only to Windows configurations in which InterSystems supplies the Telnet servers. They do not apply to UNIX® and Linux configurations, in which Telnet is supplied by the operating system vendor.

The **Telnet Settings** page (**System Administration > Configuration > Device Settings > Telnet Settings**) contains the following settings:

### DNS Lookup

On or Off. This setting enables or disables DNS lookup of the client address in the telnet daemon before passing the address to the Caché process that was created to service the connection. This determines the format of the client address returned by \$IO and \$ZIO in the Caché process. When on, a DNS lookup of the client address is performed, and the client name is passed to Caché. When off, no DNS lookup is performed, and the client address is provided in dotted decimal format. The default is true.

DNS lookup should be turned off if a DNS server is not available to do the lookup, because a long delay occurs during login if the DNS server is not available.

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

### Telnet Port Number

TCP/IP port number for Telnet connections. The default port number is 23. If multiple Caché configurations are to run on the same host at the same time, a different Telnet port number must be specified for each running configuration. Clients can attach to configurations using the non-default port number by specifying the port number when they invoke Telnet on the client system. Telnet, with or without SSL, can be configured on any port; it does not require the use of port 992.

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

## 1.4 Monitor Settings

The **Monitor Settings** page (**System Administration > Configuration > Additional Settings > Monitor**) contains the following settings:

## Start Patrol at System Startup

When you select **Yes**, the connection to PATROL starts automatically whenever Caché starts up. The default is **No**. When you edit this setting, the Caché end of the PATROL interface immediately stops and starts. For more information, see the appendix [Monitoring Caché Using BMC PATROL](#) in the *Caché Monitoring Guide*.

## Patrol Top Processes to Monitor

Number of processes displayed in the Process Status window on the PATROL console. This window shows the top processes as sorted by global or routine activity. The default number of processes is 20. A value of 0 tells the PATROL utility to stop calculating the top processes, potentially saving significant work on systems with many processes. The valid range is 1-10000 processes.

## Patrol Display Mode

Controls how the monitoring data is displayed in the PATROL console. The default option is **Total**. Options are as follows:

- **Total** displays the total counts since the collection was started.
- **Delta** displays the count for the last collection period.
- **Rate** displays a calculated count per second.

## Patrol Collection Interval Seconds

Number of seconds between each time Caché collects data and makes it available to PATROL. The default is 30 seconds; the valid range is 1-900 seconds.

## Start SNMP Agent at System Startup

To enable SNMP monitoring, select **Yes** for this setting. You must also have the %Service\_Monitor enabled on the **Services** page (**System Administration > Security > Services**). For more information, see [Monitoring Caché Using SNMP](#) in the *Caché Monitoring Guide*

## WMI Enabled

Select **Yes** to use WMI to collect Caché information. You must also enable the Caché monitoring service. For more information, see [Monitoring Caché Using WMI](#) in the *Caché Monitoring Guide*.

# 1.5 National Language Settings

The **National Language Settings** page (**System Administration > Configuration > National Language Settings**) contains the following settings:

For more information, see [Configuring NLS Settings](#) in the *Caché System Administration Guide*.

- [Locale Definitions](#)
- [Import Locals or Tables](#)

## 1.5.1 Locale Definitions

The **Locale Definitions** page (**System Administration > Configuration > National Language Settings > Locale Definitions**) contains the following settings:

## Locale Definitions

- **Validate** - Displays a message indicating the validation is successful or an appropriate error message if it is not.
- **Copy** - Enter a locale name in which to create the copy. The new locale name must contain four characters beginning with *y* and ending with *8* or *w*. The default description is *Copy of %locale*, where *%locale* is the selected locale name.
- **Export** - Enter the file name to receive the export; it must be an .xml file. The default name is *loc\_%locale.xml*, where *%locale* is the selected locale.
- **Install** - Select a locale to install that is different from the current locale. An initial validation occurs. If it fails, an error message displays; if it succeeds select **Yes - Install Now**. You can still install the locale if only a warning is displayed.
- **Load Table** - Select a table type and then a table name from the list populated after you select the type. You then may select **OK**.
- **Delete** - Disabled if the selected locale is the current locale. If you select another locale, a confirmation displays. You can select **Cancel** to stop the delete.

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

- **Basic Properties**
- **Date, Time, and Number Formats**
- **Internal Tables** - You have two options when editing the internal tables:
  - **Edit Tables** - You may select or delete a table from the list boxes by double selecting an item, or by selecting an item and then selecting the **>** or **<** to move it from the appropriate list.  
Tables that require at least one entry are indicated by an asterisk (\*); the other tables may be left empty.
  - **Edit Defaults** - You may choose the default from the values you enter in the **Edit Tables** function of the **Internal Tables** category.
- **Input/Output Tables** - You can edit, add, or remove a table when choosing to edit this category.
  - To edit a table, select the table in the first list. The table name appears in the lower box. You can modify the values and select **Save**.
  - To remove a table, select the table in the first list. The table name appears in the lower box; select **Remove**. A confirmation box displays offering you the option to **Cancel** or **OK** the delete.
  - To add a table, select **Add**. The lower box has the **Table** field enabled and the **Remove** option disabled. You can enter a table name and enter the **Output to** and **Input from** fields.

Select **Save** when you have made all your updates. If the save is successful, the updated list appears; otherwise, an appropriate error message displays.

- **Input/Output Defaults**
- **Strings**

## 1.5.2 Import Locales or Tables

The **Import Locale** page (**System Administration > Configuration > National Language Settings > Import Locales or Tables**) contains the following settings:

### Import Locale

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

## 1.6 Source Control Settings

The **Source Control Settings** page (**System Administration > Configuration > Additional Settings > Source Control**) contains the following settings. For more information on Source Control see the appendix [Using Studio Source Control Hooks in Using InterSystems Development Environments — Atelier and Studio](#).

### NAMESPACES

1. To set a source control for a namespace, select the namespace.
2. Select a source control class and select **OK**.

## 1.7 SQL and Object Settings

The **SQL and Object Settings** menu (**System Administration > Configuration > SQL and Object Settings**) contains the following settings:

- [General SQL Settings](#)
- [TSQL Compatibility Settings](#)
- [ISQL Compatibility Settings](#)
- [FileMan Conversion Settings](#)
- [System-Defined DDL Mappings](#)
- [User-Defined DDL Mappings](#)

### 1.7.1 General SQL Settings

The **General SQL Settings** page (**System Administration > Configuration > SQL and Object Settings > General SQL Settings**) contains the following settings.

Selecting a check box sets a Yes value; clearing a check box sets a No value. If you set one or more configuration options, the **General SQL Settings** label will be followed by an asterisk, indicating that changes have been made but not yet saved. You must press the **Save** button for configuration changes to take effect.

Caché includes support for the following SQL configuration settings. They are grouped by their **General SQL Settings** tab, and listed in alphabetical order.

- General SQL Settings: [SQL tab](#)
- General SQL Settings: [DDL tab](#)
- General SQL Settings: [Optimization tab](#)

Many of the same options can also be displayed by invoking `$$SYSTEM.SQL.CurrentSettings()` method.

You can also manage SQL system data types and user data types by selecting **System Administration > Configuration > SQL and Object Settings**, then either **System-defined DDL Mappings** or **User-defined DDL Mappings**.

### 1.7.1.1 SQL Tab

#### All Class Queries Project as Stored Procedures:

Yes or No. When Yes, all SQL class queries project as SQL Stored Procedures, regardless of the query's `SqlProc` value. When No, only those SQL class queries with `SqlProc=TRUE` are projected as SQL Stored Procedures. The default is No. You can also set this option using the `$$SYSTEM.SQL.SetQueryProcedures()` method.

#### Allow Extrinsic Functions in SQL Statements:

Yes or No. When Yes, extrinsic functions can be used in SQL statements through ODBC, JDBC, and Dynamic Query. The default is No. Changing this option causes the purging of all cached queries in all namespaces. You can also set this option using the `$$SYSTEM.SQL.SetAllowExtrinsicFunctions()` method.

For further details, refer to the [SELECT](#) command.

#### Cached Query - Save Source:

Yes or No. When Yes, the source code (.MAC and .INT) for cached query routines created through Dynamic SQL is saved. When No, the source code is not saved. The default is No. You can also set this option using the `$$SYSTEM.SQL.SetCachedQuerySaveSource()` method.

#### Default SQL Schema Name:

String defining the default schema name. The maximum length of the string is 128 characters. The default string is: `SQLUser`. The default schema name comes into play when an unqualified table name is encountered in an SQL statement and there is no `#import` statement specified. This setting has nothing to do with the mappings between SQL schema names and the class package name; it only specifies the default schema name. If you specify `_CURRENT_USER` as the default schema name, the default schema name becomes the username of the currently logged-in process or, if the process has not logged in, `SQLUser` becomes the default schema name. If you specify `_CURRENT_USER/name` as the default schema name, where `name` is any string of your choice, then the default schema name becomes the username of the currently logged-in process or, if the process has not logged in, `name` is used as the default schema name. For example, `_CURRENT_USER/HMO` uses `HMO` as the default schema name if the process has not logged in. Changing this option causes the purging of all cached queries in all namespaces. You can also set this option using the `$$SYSTEM.SQL.SetDefaultSchema()` method.

For further details, refer to the [CREATE TABLE](#) and [CREATE VIEW](#) commands.

#### Default time precision for GETDATE(), CURRENT\_TIME, and CURRENT\_TIMESTAMP:

Default time precision for the Time component of the value returned by the SQL scalar functions [GETDATE](#), [GETUTCDATE](#), [CURRENT\\_TIME](#), and [CURRENT\\_TIMESTAMP](#). The precision is expressed as the number of decimal digits of precision allowed for the fractional second portion of the time value. The default is 0, which

means fractional seconds are not returned in the time value. The range for this setting is 0–9 digits of precision. The actual time precision is platform dependent. You can also set this option using the `$$SYSTEM.SQL.SetDefaultTimePrecision()` method.

For further details, refer to the [SET OPTION](#) command.

### Identifier Translation - From:

String of characters that provides the “From” list for DDL Identifier Translation mappings. The maximum length of the string is 256 characters. The default string is:

```
~ `!@#%$%^&*()_+--=[ ]\{|;':",./<>?
```

These mappings filter/modify valid SQL identifier characters when translating SQL identifiers into Objects identifiers. When converting an SQL identifier to an Objects identifier at DDL runtime, the characters in the “From” string are converted to the characters in the “To” string. See Identifier Translation – To. For further details, see the “[Identifiers](#)” chapter of *Using Caché SQL*.

The “To” string of characters provides the “To” list for DDL Identifier Translation mappings. The maximum length of the string is 256 characters. The default is an empty string. You can also set this option using the `$$SYSTEM.SQL.SetDDLIdentifierTranslations()` method.

### Identifier Translation - To:

String of characters that provides the “To” list for DDL Identifier Translation mappings. The maximum length of the string is 256 characters. The default is an empty string. See Identifier Translation – From. For further details, see the “[Identifiers](#)” chapter of *Using Caché SQL*. You can also set this option using the `$$SYSTEM.SQL.SetDDLIdentifierTranslations()` method.

### Lock Threshold:

The number of SQL inserts, updates, or deletes for a single table within a single transaction that will trigger a table-level lock when reached. For example, if the lock threshold is 1000 and a process starts a transaction and then inserts 2000 rows, when the 1001st row is inserted the process will attempt to acquire a table-level lock instead of continue to lock individual rows. This is to help keep the lock table from becoming too full. The SQL lock threshold may be any integer. The default is 1000. You can also set this option using the `$$SYSTEM.SQL.SetLockThreshold()` method.

When you change and save this setting, new processes that start have the new setting.

For further details, refer to the [DELETE](#), [INSERT](#), [TRUNCATE TABLE](#), and [UPDATE](#) commands.

### Lock Timeout (in seconds):

Lock timeout for Caché locks made during execution of SQL statements. The default is 10 seconds. The range is 0–32767 seconds (up to 9 hours). Changing this option affects only new processes; existing processes are not affected. You can also set this option using the `$$SYSTEM.SQL.SetLockTimeout()` method.

For further details, refer to the [SET OPTION](#) command.

### Perform Referential Integrity Checks on Foreign Keys for INSERT, UPDATE, and DELETE:

Yes or No. When Yes, the system validates the foreign key constraint for INSERT, UPDATE, DELETE, and TRUNCATE TABLE operations. When No, Caché bypasses validation of foreign key constraints. The default is Yes. Changing this option affects only new processes; existing processes are not affected. You can also set this option using the `$$SYSTEM.SQL.SetFilerRefIntegrity()` method.

For further details, refer to the [DELETE](#), [INSERT](#), [TRUNCATE TABLE](#), and [UPDATE](#) commands.

**Retains SQL Statement as Comments in .INT Code:**

Yes or No. When Yes, embedded SQL statements are retained as comments in the .INT code version of the routine. The default is Yes. You can also set this option using the `$$SYSTEM.SQL.SetRetainSQL()` method.

**SQL Security Enabled:**

Yes or No. When Yes, all Caché SQL security is enabled. This means privilege-based table/view/procedure security is active. A user can only perform actions on a table or view for which that user has been granted privilege. When No, SQL Security is disabled. This means privilege-based table/view/procedure security is suppressed. A user can perform actions on a table or view even if that user has no privileges to do so. The default is Yes. Changing this option affects only new processes; existing processes are not affected. You can also set this option using the `$$SYSTEM.SQL.SetSQLSecurity()` method.

For further details, refer to the [GRANT](#), [CREATE TABLE](#), and [CREATE VIEW](#) commands.

**SQL SELECT Synchronizes ECP Cache:**

Yes or No. When Yes, Caché implementations that use Enterprise Cache Protocol (ECP) synchronize query results. ECP is a distributed data caching architecture that manages the distribution of data and locks among a heterogeneous network of server systems. When Yes, each time a **SELECT** statement is executed Caché forces all pending ECP requests to the database server. On completion this guarantees that the client cache is in sync. The default is No. You can also set this option using the `$$SYSTEM.SQL.SetECPSync()` method.

For further details, refer to [Queries and ECP](#) in the “Querying the Database” chapter of *Using Caché SQL*.

**Support Delimited Identifiers:**

Yes or No. When Yes, a double-quoted string ("My String") is considered a delimited identifier within an SQL statement. When No, a double-quoted string ("My String") is considered a string constant or literal string. The default is Yes. Changing this option causes the purging of all cached queries in all namespaces. You can also set this option using the `$$SYSTEM.SQL.SetDelimitedIdentifiers()` method.

For further details, refer to the [SET OPTION](#) command. For further details on [delimited identifiers](#), see the “Identifiers” chapter of *Using Caché SQL*.

**TCP Keep Alive interval (in seconds):**

An integer that specifies the number of seconds to keep alive the TCP connection. Valid values range from 1 to 432000 (432000 seconds is 5 days). The default is 300. You can also set this option using the `$$SYSTEM.SQL.SetTCPKeepAlive()` method.

For further details, refer to the [TCP Client/Server Communication](#) chapter of the *Caché I/O Device Guide*.

**1.7.1.2 DDL Tab****Allow Create Primary Key Through DDL When Key Exists:**

Yes or No. When Yes, you can create a primary key constraint to a table through DDL even when a primary key constraint already exists for that table. When No, this action triggers an error code if attempted. The default is No. You can also set this option using the `$$SYSTEM.SQL.SetDDLNo307()` method.

For further details, refer to the [CREATE TABLE](#) and [ALTER TABLE](#) commands.

**Allow DDL ADD Foreign Key Constraint when Foreign Key Exists:**

Yes or No. When Yes, you can add a foreign key through DDL even if one with the same name already exists. When No, this action triggers an error code if attempted. The default is No. You can also set this option using the `$$SYSTEM.SQL.SetDDLNo311()` method.

For further details, refer to the [ALTER TABLE](#) command.

**Allow DDL CREATE INDEX for Existing Index:**

Yes or No. When Yes, you can create an index through DDL even if one with the same name already exists. When No, this action triggers an error code if attempted. The default is No. You can also set this option using the `$$SYSTEM.SQL.SetDDLNo324()` method.

For further details, refer to the [CREATE INDEX](#) command.

**Allow DDL CREATE TABLE or CREATE VIEW for Existing Table:**

Yes or No. When Yes, you can create a table or view through DDL even if one with the same name already exists. When No, this action triggers an error code if attempted. The default is No. You can also set this option using the `$$SYSTEM.SQL.SetDDLNo201()` method.

For further details, refer to the [CREATE TABLE](#) and [CREATE VIEW](#) commands.

**Allow DDL DROP of Non-existent Index:**

Yes or No. When Yes, you can drop an index through DDL even if an index of that name does not exist. When No, this action triggers an error code if attempted. The default is No. You can also set this option using the `$$SYSTEM.SQL.SetDDLNo333()` method.

For further details, refer to the [DROP INDEX](#) command.

**Allow DDL DROP of Non-existent Table or View:**

Yes or No. When Yes, you can drop (delete) a table definition through DDL even if an index of that name does not exist. When No, this action triggers an error code if attempted. The default is No. You can also set this option using the `$$SYSTEM.SQL.SetDDLNo300()` method.

For further details, refer to the [DROP TABLE](#) and [DROP VIEW](#) commands.

**Allow DDL DROP of Nonconstraint:**

Yes or No. When Yes, you can drop a field constraint through DDL even if a constraint of that name does not exist. When No, this action triggers an error code if attempted. The default is No. You can also set this option using the `$$SYSTEM.SQL.SetDDLNo315()` method.

For further details, refer to the [ALTER TABLE](#) command.

**Are Primary Keys Created through DDL not ID Keys?:**

Yes or No. When Yes, when a Primary Key constraint is specified through DDL it does *not* also become the IDKey index in the class definition. If No, a Primary Key constraint specified through DDL also becomes the IDKey index in the class definition. The No option generally gives better performance, but means that the Primary Key fields cannot be updated. The default is Yes. You can also set this option using the `$$SYSTEM.SQL.SetDDLPrimaryKeyNotIDKey()` method.

For further details, refer to the [SET OPTION](#), [CREATE TABLE](#), and [ALTER TABLE](#) commands.

**Does DDL DROP TABLE Delete the Table's Data?:**

Yes or No. When Yes, a DDL **DROP TABLE** statement deletes the table's data. When No, it deletes the table, but does not delete the data. The default is Yes. You can also set this option using the `$$SYSTEM.SQL.SetDDLDropTabDelData()` method.

For further details, refer to the [DROP TABLE](#) command.

### 1.7.1.3 Optimization Tab

The optimization hint (%) options provide the specified optimization behavior for all queries system-wide, regardless of the presence of these optimization hint keywords in the individual queries. Corresponding keywords in individual queries are permitted but not required. You can also set these system-wide options using **SET ^%SYS("HINT","keyword")=val**. For example, **SET ^%SYS("HINT", "%NOUNIONOROPT")=1**. A system-wide hint set in this way is reflected in the Management Portal check boxes (you may need to refresh the Management Portal display by pressing the **F5** key.)

#### **DISTINCT optimization turned ON:**

Yes or No. When Yes, SQL queries involving **DISTINCT** or **GROUP BY** clauses run more efficiently by making better use of indices (if indices are available). However, the values returned by such optimized queries are collated in the same way that they are stored within the index. This means that the results of such queries may be all uppercase. This may have an effect on case-sensitive applications. The default is Yes. You can also set this option using the **SetFastDistinct()** method.

For further details, refer to the [GROUP BY](#) clause and the [DISTINCT](#) clause of the **SELECT** statement.

#### **BIAS\_QUERIES\_AS\_OUTLIER:**

Yes or No. Do not use this configuration option. Leave this check box cleared (unselected). For further details on outlier selectivity, refer to [Tune Table](#) in the “Optimizing Tables” chapter of the *Caché SQL Optimization Guide*.

#### **%ALLINDEX:**

Yes or No. When Yes, all SQL queries system-wide use all indexes that provide any benefit for the first table in the query join order. When No, the optimizer uses only those indexes that the optimizer judges to be most beneficial. The default is No. You can specify exceptions to **%ALLINDEX** in specific queries for specific conditions with the **%NOINDEX** condition-level hint. For further details, refer to the [FROM](#) clause of the **SELECT** statement.

#### **%NOFLATTEN:**

Yes or No. When Yes, all SQL queries system-wide inhibit subquery flattening. The default is No. For further details, refer to the [FROM](#) clause of the **SELECT** statement.

#### **%NOMERGE:**

Yes or No. When Yes, all SQL queries system-wide inhibit the conversion of a subquery to a view. The default is No. For further details, refer to the [FROM](#) clause of the **SELECT** statement.

#### **%NOSVSO:**

Yes or No. When Yes, all SQL queries system-wide inhibit Set-Valued Subquery Optimization (SVSO). The default is No. For further details, refer to the [FROM](#) clause of the **SELECT** statement.

#### **%NOTOPOPT:**

Yes or No. When Yes, all SQL queries system-wide disable the automatic TOP with ORDER BY optimizations. The default is No. For further details, refer to the [FROM](#) clause of the **SELECT** statement.

#### **%NOUNIONOROPT:**

Yes or No. When Yes, all SQL queries system-wide disable the automatic optimizations provided for multiple OR conditions and for subqueries against a UNION query expression. The default is No. For further details, refer to the [FROM](#) clause of the **SELECT** statement.

## 1.7.2 TSQL Compatibility Settings

The **TSQL Compatibility Settings** page (**System Administration > Configuration > SQL and Object Settings > TSQL Compatibility Settings**) contains the following settings:

### DIALECT:

The DIALECT configuration option allows you to select the Transact-SQL dialect. The available options are Sybase and MSSQL. The default is Sybase.

### ANSI-NULLS:

Specifies whether comparisons to a null value return true or false.

ON = All comparisons to a null value evaluate to Unknown.

For example: Age = Null returns false. Null is unknown, so it is false/unknown if Age = Unknown.

OFF = Comparisons of non-Unicode values to a null value evaluate to True if both values are null.

For example: Age = Null returns true for null values for Age.

### CASSEINSCOMPARE:

The CASEINSCOMPARE setting specifies non-case-sensitive equality comparisons, such as 'A'='a'. If this flag is set, the comparison operators = and <> operate without regard to case in most contexts. However, there are a few contexts where such insensitivity does not apply:

- Where a comparison is the ON condition for a JOIN.
- Where either operand is a subquery.

These exceptions exist because Caché SQL does not accept the %SQLUPPER operator in these contexts.

You can activate (=1) or deactivate (=0) CASEINSCOMPARE system-wide using the following ObjectScript command:

```
SET ^%sys( "tsql" , "CASEINSCOMPARE" )=1
```

### QUOTED\_IDENTIFIER:

The QUOTED\_IDENTIFIER configuration option allows you to select whether quoted identifiers are supported. The default is OFF (not supported).

## 1.7.3 ISQL Compatibility Settings

The **ISQL Compatibility Settings** page (**System Administration > Configuration > SQL and Object Settings > ISQL Compatibility Settings**) contains the following settings:

### Support Delimited Identifiers:

**Yes** - All ISQL identifiers are created as ordinary identifiers.

**No** - All ISQL identifiers are created as delimited identifiers.

### Generate Trace Code:

**Yes** - Trace code is generated during a session.

**No** - Trace code is not generated during a session.

**Return Result of Stored Procedure Call as Resultset:**

**Yes** - Return result of a stored procedure call as a resultset.

**No** -

**Reserved Word Prefix:**

Specify a prefix to be appended to a column name when that column name conflicts with an SQL reserved word.

## 1.7.4 FileMan Conversion Settings

The **FileMan Conversion Settings** page (**System Administration > Configuration > SQL and Object Settings > FileMan Conversion Settings**) contains the following settings. These settings define how FileMan data structures are converted to InterSystems class definitions.

**Owner of the Classes Created:**

For each class, the class keyword **Owner** is created with the value specified in the settings. The default is the current value in *\$Username*. The *owner* of the classes is granted all SQL privileges on all SQL objects projected by the class (tables, views, stored procedures). The owner can insert, update, delete, and execute statements against the projected tables, views, and procedures.

**Package Name to Create the Classes in:**

Package name to create the classes in.

**Note:** If you previously mapped to one package and map the FileMan files again to a new package, the old classes are not automatically deleted.

**SuperClasses:**

A string of comma-delimited classes to which each mapped class should extend.

**Table Name Format Based on the File Name and Number:**

Specifies the format of the generated table name. For this setting, specify a string that uses the following keywords along with any characters that are valid to use as table names:

- <FILENAME> — Replaced with the name of the FileMan file
- <FILENUMBER> — Replaced with the file number

If you use <FILENAME>, any decimal place characters in the file number are converted to underscore characters.

See the examples for **Child Table Name Format Based on the File Name and Number**.

**Child Table Name Format Based on the File Name and Number:**

Specifies the format of the generated child table names. For this setting, specify a string that uses the following keywords along with any characters that are valid to use as table names:

- <FILENAME> — Replaced with the name of the FileMan file
- <FILENUMBER> — Replaced with the file number
- <PARFILENAME> — Replaced with the name of the parent file
- <PARFILENUMBER> — Replaced with the file number of the parent file

If you use <FILENAME> or <PARFILENUMBER>, any decimal place characters in the file number are converted to underscore characters.

Some examples of this setting:

- SUB\_<FILENAME> — In this example, the table name of a child table will be the string SUB\_ followed by the name of the file. For example: SUB\_ACCESSIBLE\_FILE
- f<PARFILENUMBER>c<FILENUMBER> — In this example, the table name of a child table will be the string f, followed by the number of the parent file, followed by c, followed by the number of this file. For example: f200c200\_032
- <FILENAME> — In this example, the child table name is simply the same as the filename.

### Datatype to use for FileMan Date fields:

Specifies the data type to use when mapping FileMan DATE fields. The default is %Library.FilemanDate.

### Datatype to use for FileMan DateTime fields:

Specifies the data type to use when mapping FileMan DATE/TIME fields. The default is %Library.FilemanTimeStamp.

### Expand Pointers?

Specifies whether the utility creates additional computed properties to expand the pointer field.

If it does so, the utility creates a computed property that expands the pointer field and is equal to the NAME (.01) field in the referenced file.

### Extended Mapping:

Specifies a string to be inserted into the global name in the map definitions for extended mapping purposes. For example, you might specify [ "SD" ] and your global will be mapped as ^[ "SD" ]LR( . . . ) instead of ^LR( . . . ).

This can be any valid string that can be used for extended global mapping and should include the [ . . . ] or | . . . | brackets.

### Name of the IEN Field:

Specifies the name of the IEN field, which is IEN by default.

### Retain Class?

Specifies whether to recreate the entire class if it already exists.

- **No** - The utility deletes and recreates the class, which means that SQL privileges and any add-ons to the class will be lost.
- **Yes** - The utility recreates the properties, storage, indexes, foreign keys, and so on, rather than the entire class.

### Recursion:

Controls whether sub-files and pointer are also mapped:

- **No recursion** - only this file is mapped. No sub-files or pointers are mapped.
- **Partial recursion** - the file is mapped, along with one level of sub-files and pointers.
- **Full recursion** - the file is mapped, along with all sub-files and pointers. This is the default.

**Word-Processing Fields Conversion:**

Specifies how to map word-processing fields. Select either **Convert as child tables** or **Convert as list collections**.

**Read Only?Log File:**

Specifies whether the generated classes should be read-only.

**Compile Classes:**

Specifies whether to compile the classes after creating them.

**Compile flags:**

Specifies any class compiler qualifiers and/or flags. The default is `/display=all/lock=0`

**Delete Flags:**

Specifies any class deletion qualifiers and/or flags. Default is `/display=all`

**Display Result:**

Controls how the results are displayed, **No screen display**, **Minimal screen display**, **Full screen display**.

## 1.7.5 System-Defined DDL Mappings

The **System-Defined DDL Mappings** page (**System Administration > Configuration > SQL and Object Settings > System-Defined DDL Mappings**) contains the following settings:

**System-Defined DDL Mappings**

Collection of system-defined datatype descriptions. Each description maps an SQL datatype to its Caché equivalent. See [UserDataTypes](#).

**Configuration Fields**

**Key**-the name of the SQL datatype plus any allowed arguments.

**Value**-the Caché equivalent, including any constraints on the arguments

## 1.7.6 User-Defined DDL Mappings

The **User-Defined DDL Mappings** page (**System Administration > Configuration > SQL and Object Settings > User-Defined DDL Mappings**) contains the following settings:

**User-Defined DDL Mappings**

Collection of user-defined datatype descriptions. Each description maps an SQL datatype to its Caché equivalent. See [SystemDataTypes](#).

**Configuration Fields**

**Key**-the name of the SQL datatype plus any allowed arguments.

**Value**-the Caché equivalent, including any constraints on the arguments

## 1.8 Startup Settings

The **Startup Settings** page (**System Administration > Configuration > Additional Settings > Startup**) contains the following settings:

### CallinHalt

True or false. When true, Caché executes the CALLIN^%ZSTOP routine entry each time an external program ends a CALLIN. The default is true.

### CallinStart

True or false. When true, Caché executes the CALLIN^%ZSTART routine entry each time an external program initiates a CALLIN. The default is true.

### ClisysName

Node name for the local system. The value *n* has many uses:

- The node name to be sent to the ECP network server, so that the server can identify the client
- The node name for a unique \$JOB value. This is useful when using \$JOB to index globals accessed by more than one networked system.
- Certain forms of the \$SYSTEM function return this node name concatenated with the Caché instance name, as *nodename:instancename*. This concatenated string is recorded in Audit files.

The default is blank. If no name is provided, Caché reads the computer settings and uses the computer “host name” as the client node name. This can be up to 64 characters long.

### DBSizesAllowed

Comma-separated list of database block sizes (in multiples of 1024) allowed to be entered in the database creation screen. Default is 8192 (you cannot clear this check box). Example: 8192,16384. It is important to note that when you add a database block size, you *must* also allocate memory to the database cache for that size in order to create global buffers for the block size and enable database creation; for more information, see [Allocating Memory to the Routine and Database Caches](#) in the “Preparing to Install” chapter of the *Caché Installation Guide*. For guidelines for choosing the appropriate database block/global buffer sizes for your applications, see [Large Block Size Considerations](#) in the “Configuring Caché” chapter of the *Caché System Administration Guide*.

### DefaultPortBindAddress

Specify an IP address for the SuperServer to bind to.

Set to one of the host system's IP addresses. The SuperServer binds to that address. Requests to the SuperServer port on other IP addresses on the host are not accepted. This makes it possible to limit connections to the SuperServer to a single address on a multihomed host. The SuperServer is the process that accepts client connections for Cache Direct, ODBC, JDBC, Cache Studio, and other connection technologies. If this property is not set, the SuperServer accepts requests on all IP addresses on the host. The default is to accept on all addresses.

### EnableVSSBackup

True or false. When true, Caché supports the Volume Shadow Copy Service (VSS) on Windows. See the [Backup Strategies](#) section in the “Backup and Restore” chapter of *Caché Data Integrity Guide* for information about creating a backup using VSS or other methods.

## EnsembleAutoStart

*Ensemble only.*

True or false. When true, the production you set to auto-start in each Ensemble namespace starts when you start Ensemble. To facilitate debugging situations involving troubled productions, you can disable this setting to prevent a production from starting. This entry is ignored if this is not an Ensemble system. The default is true.

See the description of the Auto-Start Production field in the chapter “Starting and Stopping Productions” of *Managing Ensemble Productions* for details on how this setting works with Ensemble production settings.

## ErrorPurge

Number of days before the error globals for the `^%ETN` error handler are marked for purge. The purge occurs at the next Caché startup. The default is 30 days. The range is 1–1000 days.

If you edit this setting, the change is applied the next time you restart Caché.

## FIPSMode

True or false. When true, Caché uses the FIPS 140–2 compliant library for database encryption on Red Hat Enterprise Linux 6.6 (or later minor version) and Red Hat Enterprise Linux 7.1 (or later minor version) for x86-64. The default is false. See the article [FIPS 140–2 Compliance for Caché Database Encryption](#) for details.

## IPv6

True or false. When true, this indicates that your system is operating in an IPv6 (Internet Protocol Version 6) network with IPv6 addresses. The default is false.

## JobHalt

True or false. When true, Caché executes the `JOB^%ZSTOP` routine entry each time a background process ends. The default is true. Background processes include any processes that are started via the `JOB` command, plus any background server processes including CSP, Caché Direct, ODBC, or any of the objects bindings.

## JobServers

Number of job servers you want Caché to start up. The default is 0. The range is 0–2000.

Having a large number of job servers running will use more memory and processes, but allows for much faster jobbing of processes because Caché doesn't have to start the processes at the system level and then initialize them.

Job servers are best used when the application creates a significant number of short-lived processes via the `Job` command. For this type of process where operating system process creation overhead dominates the total cost of running the process, using job servers can be beneficial. If background processes tend to perform extended tasks then there is very little benefit from using job servers.

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

## JobStart

True or false. When true, Caché executes the `JOB^%ZSTART` routine entry each time a background process starts up. The default is true. Background processes include any processes that are started via the `JOB` command, plus any background server processes including CSP, Caché Direct, ODBC, or any of the objects bindings.

## MaxCacheTempSizeAtStart

Maximum size in megabytes of the CacheTemp database when the system is restarted. When the system restarts, the CacheTemp database is truncated to this size. If 0, the CacheTemp database is not truncated. The range is 0-1000000.

**MaxConsoleLogSize**

Maximum size of the Caché console file, in megabytes. The console file is either the `cconsole.log` in the Caché system management directory, or the `filename.log` named by the [ConsoleFile](#) setting.

If you enter a value (in MB) that is smaller than the current setting of `MaxConsoleLogSize`, then the current console log file, `filename.log`, is immediately renamed to `filename.old_Date`. The system creates a new `filename.log` file. New entries are appended to the newly-created file.

If the console file grows larger than the size set with `MaxConsoleLogSize`, the task manager `Purge errors and log files` task performs the rename and the `filename.log` creation the next time it runs.

The default `ConsoleLogSize` is 5 megabytes. The range is 1–500 megabytes.

**ProcessHalt**

True or false. When true, Caché executes the `LOGIN^%ZSTOP` routine entry each time a terminal user logs out. The default is true.

**ProcessStart**

True or false. When true, Caché executes the `LOGIN^%ZSTART` routine entry each time a terminal user logs in. The default is true.

**ShutdownTimeout**

Number of seconds Caché should wait for shutdown to complete normally before timing out and forcing a shutdown. The default is 300 seconds (5 minutes). The range is 120 to a maximum of 100,000 seconds.

If you edit this setting, the change applies automatically the next time you restart Caché.

**SystemHalt**

True or false. When true, Caché executes the `SYSTEM^%ZSTOP` routine entry each time it shuts down. The default is true.

**SystemStart**

True or false. When true, Caché executes the `SYSTEM^%ZSTART` routine entry each time it starts up. The default is true.

If you edit this setting, the change applies automatically the next time you restart Caché.

**TempDirectory**

Name of a subdirectory in which to store temporary files for Caché. The string is assumed to be a relative pathname, the name of a subdirectory under the `Mgr` subdirectory in the Caché installation directory. The default value for `TempDirectory` is:

`Temp`

Which identifies the following location under `install-dir`, the Caché installation directory:

`install-dir\Mgr\Temp`

When you set a new `TempDirectory` value, the system creates a subdirectory of this name under the `Mgr` subdirectory in the Caché installation directory, as shown for `Temp` in the previous example. This new directory becomes the Caché temporary directory.

## TerminalPrompt

A comma separated string of values (0–8) which set the default terminal prompt for the system. The default is 8,2

The order of the values in the string is the order the values appear in the prompt. For example

```
TerminalPrompt="2,1"
```

gives you a terminal prompt of %SYS:HostName>

- 0 - Use only ">" for the prompt.
- 1 - Host name, also known as the current system name. The name assigned to your computer. For example, LABLAPTOP>. This is the same for all of your terminal processes.
- 2 - Namespace name. For example, %SYS>. The current namespace name is contained in the \$NAMESPACE special variable. It can be an explicit namespace name or an implied namespace name.
- 3 - Config name. The name of your system installation. For example, CACHE2>. This is the same for all of your terminal processes.
- 4 - Current time, expressed as local time in 24-hour format with whole seconds. For example, 15:59:36>. This is the static time value for when the prompt was returned. This value changes for each prompt.
- 5 - pid. The Process ID for your terminal. For example, 2336>. This is different for each terminal process. This value can also be returned from the \$JOB special variable.
- 6 - Username. For example, fred>. This is the same for all of your terminal processes.
- 7 - Elapsed time executing the last command, in seconds.milliseconds. For example, .000495>. Leading and trailing zeros are suppressed. This changes for each prompt.
- 8 - Transaction Level. For example, TL1>.

## WebServer

Start the private Webserver when Caché starts. For information on the private Webserver, see the section [Minimal Apache Web Server or Private Web Server](#) in the [CSP Gateway Configuration Guide](#).

- True - When Caché starts, the private Webserver is started.
- False - When Caché starts, the private Webserver is not started.

## WebServerName

The DNS name or IP address of the Web Server that is configured for use with Caché tools. For example if Studio is connected to this Caché system, and if the user requests to run a CSP template from Studio, WebServerName identifies the DNS name or IP address to which this request will be sent. See [WebServerPort](#).

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

## WebServerPort

Web Server port number. Setting this number to any non-zero value enables the Web Server on that port. If you set this number to 0, the Web Server is disabled. A standard Caché installation sets the Web server port number to the first unused port number greater than or equal to 57772. The Web server runs on the same system as the Caché server.

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

**WebServerURLPrefix**

The Caché instance name. This is the name you assigned to Caché when you installed it. You must provide the WebServerURLPrefix value for each Caché instance if you are using a remote Web server to access one or more Caché instances.

This is only one of the steps required to set up a remote Web server to access one or more Caché instances. For details, see the “[Connecting to Remote Servers](#)” chapter in the *Caché System Administration Guide*.

**ZSTU**

True or false. When true, Caché runs the user-defined startup from the ^ZSTU routine. The default is true.

If you edit this setting, the change applies automatically the next time you restart Caché.

## 1.9 Task Manager Email Settings

The **Task Manager Email Settings** page (**System Administration > Configuration > Additional Settings > Task Manager Email**) contains the following settings. For more information, see [Configuring Task Manager Email Settings](#) in the *Caché System Administration Guide*.

**SMTP Server and Port**

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

**SSL Config**

The SSL configuration to use if you want to encrypt the email using SSL/TLS. If there are no SSL configurations on the instance, or you want to create a new one, see [Creating or Editing an SSL/TLS Configuration](#) in the “Using SSL/TLS with Caché” chapter of the *Caché Security Administration Guide*. If you do not select an SSL configuration, SSL/TLS will not be used.

**SMTP Auth User and Password**

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

**Sender**

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

**Reply To**

Email address to which the recipient should reply

**Success Subject**

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

**Success Message**

The formatted message sent after a successful task runs

**Failure Subject**

The formatted subject line of a failed task message

## Failure Message

The formatted message sent after a task fails

## Parameters for Subjects and Messages

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

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

# 1.10 Zen Reports

The following settings are on the **System Administration > Configuration > Zen Reports** menu:

- [Zen Report Render Servers](#)
- [Zen Report Settings](#)

## 1.10.1 Zen Report Render Servers

The **Zen Reports Render Servers** page (**System Administration > Configuration > Zen Reports > Render Servers**) contains the following settings:

### New Zen Report Server

- **Name:** A unique name for the Render Server.
- **Port:** The TCP port that the Render Server uses to receive reports to render.
- **Ping Port:** The TCP port that the Render Server uses for all other communication, such as status queries and shutdown requests.
- **Num Threads:** If the Render Server is using multi-threaded Java, this field supplies the number of threads used by the Render Server for report rendering.

- **Num Ping Threads:** If the Render Server is using multi-threaded Java, this field supplies the number of threads used by the Render Server for other communication.
- **PDF Renderer:** The renderer used by the Render Server for PDF rendering of Zen reports. The value FOP is the default and refers to the version of Apache FOP that is installed with Caché. If you select RenderX XEP, you must specify the location of the configuration file, `xep.xml` by providing an `XEP_HOME` environment variable.
- **Renderer Configuration File:** A file that contains configuration information for the built-in FOP renderer. This field is automatically filled with the name of the default file, `C:\MyCache\fop\conf\fop.xconf`. This file is supplied with the built-in FOP. You can use it as a template for a custom file. The file `C:\MyCache\fop\conf\fop.xconf_dist` is a backup copy. If you want to use a different configuration file, provide the file name here.

This field does not appear on the form if you select the RenderX XEP PDF renderer. You must use the `XEP_HOME` environment variable to specify the location of the `xep.xml`, the RenderX XEP configuration file.

- **Log Level:** Standard Java parameters to control logging. If you choose to enable logging, the following three items appear on the form:

- **Log File:** By default, the Render Server log file is created in your home directory. You can specify an alternate location here.

Each time you stop and restart the Render Server, it begins a new log file. The Render Server also starts a new log file when the file size exceeds the limit set by **Max. File Size**. The log file names have a numeric suffix. The file ending in `.0` is the most recent, and as new files are created, the previous ones are renamed with larger suffix numbers, until the number of files reaches the limit set by **Rotation Count**. Then the names recycled and older information is lost. This field supplies the path and base file name of the log file.

If you configure more than one Render Server, providing log file names makes it easy for you to match log files with the Render Server that created them.

- **Max. File Size:** Maximum size of the Render Server log file. The Render Server creates a new log file when the size of the current log file reaches this limit.
- **Rotation Count:** The maximum number of log files. The Render Server recycles file names, losing older information, when the number of log files reaches this limit.

- **Initialization Timeout:** The amount of time in seconds that Zen reports waits for the Render Server to start up. An error occurs if the Render Server fails to start in this time.
- **Connection Timeout:** The amount of time in seconds that Zen reports waits for the Render Server to connect when rendering a report. You normally expect connection to take less time than initialization. An error occurs if the Render Server fails to connect in this time.
- **JavaHome:** The location where Java is installed. Used if you have more than one installation of Java.
- **JVM Additional Args:** Any additional arguments you want to send to the Java Virtual Machine. This feature is not implemented at this time.

Enter the full path to the `fop` or the `xep` executable file and select **Save**. For windows, `fop.bat` or `xep.bat`; for UNIX®, `fop.sh` or `xep.sh`. For more information, see the section [Configuring Zen for PDF Output](#) in the book *Using Zen Reports*.

If you configure a Render Server to use RenderX XEP, two additional fields appear on the configuration page:

- **How Often To Clean (XEP):** The interval in seconds the RenderServer uses to check whether RenderX has processed enough files to require cleaning. The default value is 300 seconds (5 minutes).

- **Num. Files Before Clean (XEP):** The number of files RenderX can process before the Render Server initiates a cleaning operation. The default value is 100.

RenderX can consume memory as it runs, so a cleaning operation needs to be performed periodically. Cleaning also consumes resources, so these two fields are provided to let you set the parameters that determine when cleaning takes place. The need for cleaning is determined by the number of files RenderX has processed. Use the field **Num. Files Before Clean (XEP)** to set this number. The value set in **How Often To Clean (XEP)** determines how often the Render Server checks whether RenderX has reached the file limit.

Once you have saved your changes, use the **Cancel** button to return to the **Render Servers** page (**System Administration > Configuration > Zen Reports > Render Servers**), where you see that the new Render Server has been added to the list.

## 1.10.2 Zen Report Settings

The **Zen Report Settings** page (**System Administration > Configuration > Zen Reports > Settings**) contains the following settings:

### Path and file name to be used for PDF generation:

Enter the full path to the `fop` or the `xep` executable file and select **Save**. For windows, `fop.bat` or `xep.bat`; for UNIX®, `fop.sh` or `xep.sh`. For more information, see the section [Configuring Zen for PDF Output](#) in the book *Using Zen Reports*.

### Configuration file:

Enter the full path name of a configuration file to use for Zen Reports and select **Save**. If this field is empty and Zen Reports does not set `USEINSTALLEDFOF=0`, the file `fop.xconf` is used.

### Adobe Path for pdfprint:

Enter or select the full path to the Adobe Reader and select **Save**.

### Java\_Home environment variable:

The Java home environment variable is displayed here if one is set.

### To verify that the xsl engine is configured correctly, select here: **Verify Now**

Select **Verify Now** to ensure that you have installed either the `fop` or `xep` programs and that the accurate full path to the executable file (appropriate to your operating system) is named and (was saved) in the file name field below. For more information, see the section [Configuring Zen for PDF Output](#) in the book *Using Zen Reports*.

