



# Configuration Parameter File Reference

Version 2019.4  
2020-01-28

*Configuration Parameter File Reference*

InterSystems IRIS Data Platform Version 2019.4 2020-01-28

Copyright © 2020 InterSystems Corporation

All rights reserved.

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

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

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

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

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

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

For Support questions about any InterSystems products, contact:

**InterSystems Worldwide Response Center (WRC)**

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: support@InterSystems.com

# Table of Contents

<b>About This Book .....</b>	<b>1</b>
<b>1 Introduction to the Configuration Parameter File .....</b>	<b>3</b>
1.1 CPF Overview .....	3
1.1.1 CPF Format .....	3
1.2 Editing the Active CPF .....	4
1.2.1 Choosing a CPF on Startup .....	5
1.2.2 Defining a CPF in a Deployment .....	5
1.3 Configuration Security .....	6
1.4 Parameter Descriptions .....	6
<b>[ComPorts] .....</b>	<b>7</b>
COMn .....	8
<b>[config] .....</b>	<b>11</b>
LibPath .....	12
MaxServerConn .....	13
MaxServers .....	14
Path .....	15
bbsiz .....	16
console .....	17
errlog .....	18
globals .....	19
gmheap .....	20
history .....	21
ijcbuf .....	22
ijcnum .....	23
jrnbufs .....	24
locksiz .....	25
memlock .....	26
netjob .....	28
nlstab .....	29
overview .....	30
pijdir .....	31
routines .....	32
udevtabsiz .....	34
wijdir .....	35
targwijsz .....	36
zfheap .....	37
<b>[ConfigFile] .....</b>	<b>39</b>
Version .....	40
<b>[Databases] .....</b>	<b>41</b>
Database .....	42
<b>[Debug] .....</b>	<b>45</b>
Dumpstyle .....	46
Semserset .....	47
<b>[Devices] .....</b>	<b>49</b>

Devices .....	50
<b>[DeviceSubTypes] .....</b>	<b>53</b>
DeviceSubTypes .....	54
<b>[ECP] .....</b>	<b>57</b>
ClientReconnectDuration .....	58
ClientReconnectInterval .....	59
ServerTroubleDuration .....	60
<b>[ECPServers] .....</b>	<b>61</b>
ECPServers .....	62
<b>[IO] .....</b>	<b>63</b>
Other .....	64
File .....	65
Terminal .....	66
<b>[Journal] .....</b>	<b>67</b>
AlternateDirectory .....	68
BackupsBeforePurge .....	69
CurrentDirectory .....	70
DaysBeforePurge .....	71
FileSizeLimit .....	72
FreezeOnError .....	73
JournalFilePrefix .....	74
JournalcspSession .....	75
<b>[LicenseServers] .....</b>	<b>77</b>
LicenseServers .....	78
<b>[Map] .....</b>	<b>79</b>
Global .....	80
Package .....	82
Routine .....	83
<b>[MapMirrors] .....</b>	<b>85</b>
MapMirrors .....	86
<b>[MirrorMember] .....</b>	<b>89</b>
AgentAddress .....	90
AsyncMemberGUID .....	91
AsyncMemberType .....	92
AsyncUseSystemPurgeInterval .....	93
JoinMirror .....	94
SystemName .....	95
ValidatedMember .....	96
VirtualAddressInterface .....	97
<b>[Mirrors] .....</b>	<b>99</b>
Mirrors .....	100
<b>[Miscellaneous] .....</b>	<b>103</b>
AsyncDisconnectErr .....	104
AsynchError .....	105
BreakMode .....	106
CollectResourceStats .....	107

DisconnectErr .....	108
FileMode .....	109
GlobalKillEnabled .....	110
IEEEError .....	111
LineRecall .....	112
LogRollback .....	113
MVDefined .....	114
NodeNameInPid .....	115
NullSubscripts .....	116
OldZU5 .....	117
OpenMode .....	118
PopError .....	119
RefInKind .....	120
ScientificNotation .....	121
SetZEOF .....	122
ShutDownLogErrors .....	123
StopID .....	124
SwitchOSdir .....	125
SynchCommit .....	126
TelnetNUL .....	127
TruncateOverflow .....	128
Undefined .....	129
UseNagleAlgorithm .....	130
ViewPastData .....	131
ZDateNull .....	132
<b>[Monitor]</b> .....	<b>133</b>
SNMPEnabled .....	134
<b>[Namespaces]</b> .....	<b>135</b>
Namespace .....	136
<b>[SQL]</b> .....	<b>139</b>
AllowRowIDUpdate .....	140
ANSIPrecedence .....	141
AutoParallel .....	142
AutoParallelThreshold .....	143
BiasQueriesAsOutlier .....	144
Comment .....	145
DBMSSecurity .....	146
DDLDefineBitmapExtent .....	147
DDLFinal .....	148
DDLNo201 .....	149
DDLNo30 .....	150
DDLNo307 .....	151
DDLNo311 .....	152
DDLNo315 .....	153
DDLNo324 .....	154
DDLNo333 .....	155
DDLSQLOnlyCompile .....	156
DDLUseExtentSet .....	157
DDLUseSequence .....	158

DefaultSchema .....	159
DelimitedIds .....	160
DropDelete .....	161
ECPSync .....	162
ExtrinsicFunctions .....	163
FastDistinct .....	164
IdKey .....	165
IdTrxFrom .....	166
IdTrxTo .....	167
JavaClassPath .....	168
JavaHome .....	169
JDBCGatewayJVMArgs .....	170
JDBCGatewayLog .....	171
JDBCGatewayPort .....	172
JDBCGatewayUsePassphrase .....	173
LockThreshold .....	174
LockTimeout .....	175
ODBCVarcharMaxlen .....	176
QueryProcedures .....	177
RPC .....	178
ReferentialChecks .....	179
SaveMAC .....	180
TCPKeepAlive .....	181
TimePrecision .....	182
TODATEDefaultFormat .....	183
<b>[SqlSysDatatypes]</b> .....	<b>185</b>
System Datatypes .....	186
<b>[SqlUserDatatypes]</b> .....	<b>187</b>
User Datatypes .....	188
<b>[Startup]</b> .....	<b>189</b>
CallinHalt .....	190
CallinStart .....	191
CliSysName .....	192
DBSizesAllowed .....	193
DefaultPort .....	194
DefaultPortBindAddress .....	195
EnableVSSBackup .....	196
EnsembleAutoStart .....	197
ErrorPurge .....	198
FIPSMode .....	199
IPv6 .....	200
JobHalt .....	201
JobServers .....	202
JobStart .....	203
LicenseID .....	204
MaxConsoleLogSize .....	205
MaxIRISTempSizeAtStart .....	206
ProcessHalt .....	207
ProcessStart .....	208

ShutdownTimeout .....	209
SystemHalt .....	210
SystemStart .....	211
TempDirectory .....	212
TerminalPrompt .....	213
WebServer .....	214
WebServerName .....	215
WebServerPort .....	216
WebServerURLPrefix .....	217
ZSTU .....	218
<b>[Telnet] .....</b>	<b>219</b>
DNSLookup .....	220
Port .....	221
<b>Appendix A: CPF Settings in the Management Portal .....</b>	<b>223</b>
A.1 SQL and Object Settings .....	223
A.1.1 SQL .....	223
A.1.2 Objects .....	225
A.1.3 TSQL Compatibility .....	225
A.1.4 ISQL Compatibility .....	225
A.1.5 System DDL Mappings .....	225
A.1.6 User DDL Mappings .....	225
A.2 National Language Settings .....	226
A.3 Device Settings .....	226
A.3.1 Devices .....	226
A.3.2 Device Subtypes .....	226
A.3.3 IO Settings .....	226
A.3.4 Telnet Settings .....	226
A.4 Additional Settings .....	227
A.4.1 Compatibility Settings .....	227
A.4.2 Advanced Memory .....	229
A.4.3 Monitor Settings .....	231
A.4.4 Source Control .....	231
A.4.5 Startup Settings .....	231
A.4.6 Task Manager Email .....	234



# About This Book

This book describes the settings in the *configuration parameter file*, iris.cpf, also called the CPF, which determine the InterSystems IRIS® data platform environment.

This book contains the following chapters. Except for the introduction, each section in the book is a reference that corresponds to a section of the CPF.

- [Introduction to the Configuration Parameter File](#)
- [ComPorts](#)
- [config](#)
- [ConfigFile](#)
- [Databases](#)
- [Debug](#)
- [Devices](#)
- [DeviceSubTypes](#)
- [ECP](#)
- [ECPServers](#)
- [IO](#)
- [Journal](#)
- [LicenseServers](#)
- [Map](#)
- [MapMirrors](#)
- [MirrorMember](#)
- [Mirrors](#)
- [Miscellaneous](#)
- [Monitor](#)
- [Namespaces](#)
- [SQL](#)
- [SqlSysDatatypes](#)
- [SqlUserDatatypes](#)
- [Startup](#)
- [Telnet](#)

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

The appendix of this book, [CPF Settings in the Management Portal](#), describes how to modify the CPF using the Management Portal. For information on the other pages in the Management Portal, see the [Management Portal Page Reference](#) section of the “Using the Management Portal” chapter in *System Administration Guide*.



# 1

# Introduction to the Configuration Parameter File

When InterSystems IRIS® data platform starts, it reads configuration values from the text file `iris.cpf`, the *configuration parameter file* (CPF). This file defines the particular InterSystems IRIS configuration for each instance. This chapter describes:

- [CPF Overview](#) – A description of the CPF and its syntax.
- [Editing the Active CPF](#) – The various ways to modify the CPF.
- [Configuration Security](#) – How to password-protect the CPF against unwanted changes.
- [Parameter Descriptions](#) – The typical structure of entries this book.

The “[Table of Contents](#)” at the beginning of this book shows a brief description of every parameter in the CPF. The order of the “Table of Contents” matches how parameters are ordered within the CPF.

**Note:** This document and the CPF both refer to memory amounts as powers of two. For example, a kilobyte (KB) means 1024 bytes, and a megabyte (MB) means 1024 KB.

## 1.1 CPF Overview

The configuration parameter file, also called the CPF, defines an *InterSystems IRIS Configuration*. On startup, InterSystems IRIS reads the CPF to obtain the values for most of its settings.

The default CPF, `iris.cpf`, is located in the [installation directory](#). There are multiple ways to modify the CPF, as described in the [Editing the Active CPF](#) section of this chapter.

InterSystems IRIS creates multiple backups of the CPF. Once per day, if the `iris.cpf` file is modified, InterSystems IRIS creates a backup in the same directory called `iris.cpf_yyyymmdd`. These backups are automatically purged after one year. Additionally, after a successful startup or shutdown, a copy of the CPF is saved in the installation directory as `_LastGood_.cpf`, which represents the most recent valid configuration.

### 1.1.1 CPF Format

A configuration parameter file is a line-oriented, UTF-8 text file with a `.cpf` extension. Each line ends with a carriage return and line feed. Long items cannot be continued on a following line. A line in the file is one of the following elements:

- **Blank Space** – An empty line made up of zero or more spaces.
- **Section Heading** – The name of a file section enclosed in square brackets [].
- **Parameter** – An InterSystems IRIS configuration parameter and its value(s).
- **Comment** – A comment added by a user.

### 1.1.1.1 Blank Space

In general, spaces at the beginning and end of lines are without effect. Spaces within the line are usually considered significant. The best practice is to use no spaces in the line except where they are meaningful components of strings.

### 1.1.1.2 Section Headings

Related settings are collected into sections. The beginning of a section is marked by a line consisting of the name of the section enclosed in square brackets. For example:

```
[Devices]
```

All lines after the section heading, up to the next section name (or the end of file), are in the same section.

### 1.1.1.3 Parameters

Each line beneath a section heading is the definition of a parameter. Each parameter line uses the following syntax, where *keyword* is a parameter name and *value* is a string:

```
keyword=value
```

When there is a set of similar items to configure, they may be displayed as *keyword\_#*. Examples include namespaces, databases, devices, and anything else of which there is a group or set of similar items to configure, one per line. The syntax is:

```
keyword_1=value  
keyword_2=value  
keyword_n=value
```

The syntax for the *value* string varies widely from parameter to parameter. The string may indicate true or false using 1 or 0; it may be a number of bytes, or a number of megabytes; it may be a single value, or it may contain multiple values separated by a delimiter character on the same line. If there is a delimiter within the string, it may be a comma, semicolon, tilde (~), slash (/), colon, or some combination of these, depending on the parameter.

### 1.1.1.4 Comments

The CPF supports comments. These can appear on a single line or across multiple lines. Comments can start at the beginning of the line or after other content on a line.

To introduce a single-line comment use ";" (semicolon), "#" (pound sign), or "://" (two slashes).

To introduce a multiline comment, use "/\*" (slash, asterisk) to begin the command and "\*/" (asterisk, slash) to end it.

## 1.2 Editing the Active CPF

There are multiple ways to interact with the CPF, including through the Management Portal, API calls, or a text editor. For instructions on how to change a specific parameter, review the *Changing This Parameter* section of the reference page for that parameter. Some changes may require the instance to be restarted to take effect.

When using a text editor to modify the CPF, you must first shut down the instance. Be sure to follow the syntax described in the [CPF Format](#) section of this chapter. InterSystems recommends that you save a backup copy of the CPF before editing it, as a CPF with improper formatting or values may cause InterSystems IRIS to fail to start.

You can specify the CPF for InterSystems IRIS to use with the **iris start** command, or you can create a partial CPF to merge into iris.cpf during deployment on UNIX® and Linux systems. These options are described in the following sections:

- [Choose a CPF on Startup](#)
- [Defining a CPF in a Deployment](#)

## 1.2.1 Choosing a CPF on Startup

If you frequently switch between two or more InterSystems IRIS configurations, such as for development and testing purposes, you can create distinct CPFs for these purposes. When starting InterSystems IRIS, you can specify which .cpf file to use to reduce time spent manually changing settings.

For example, on Windows, if the InterSystems IRIS installation directory is C:\MyIRIS, you might have the following CPFs:

```
C:\MyIRIS\iris.cpf           ; default CPF
C:\MyIRIS\production.cpf    ; for production
C:\MyIRIS\development.cpf   ; for development
C:\MyIRIS\testapps.cpf      ; for testing
C:\MyIRIS\iris_customerbug.cpf ; for troubleshooting
```

To use a different CPF, you must first stop InterSystems IRIS. Then, start the instance with the **iris start** command, specifying the full path of the CPF InterSystems IRIS should use. The **iris start** command is described in the [Controlling an InterSystems IRIS Instance](#) section of the “Using Multiple Instances of InterSystems IRIS” chapter in *System Administration Guide*.

At shutdown, the system automatically saves the last known error-free configuration to a file called \_LastGood\_.cpf in the installation directory. You can use this file, if you need to, for recovery purposes.

## 1.2.2 Defining a CPF in a Deployment

On UNIX® and Linux, if you want to modify the default CPF that is deployed with an instance of InterSystems IRIS, you can declare the desired settings in a partial CPF called a *merge file*. When deploying an instance, you can set the environment variable **ISC\_CPF\_MERGE\_FILE** to point to the merge file. Before starting up for the first time, InterSystems IRIS updates the default CPF with the values from the merge file. This allows you to deploy multiple instances from the same source, each with a customized CPF.

You can specify a merge file when performing a silent installation, deploying with InterSystems Cloud Manager (ICM), or creating and starting a container on your own. A merge file can have any name, for example custom-params.config, and is syntactically identical to a CPF, with each parameter occupying a single line beneath the appropriate section heading, as described in the [CPF Format](#) section in this chapter. The merge file can contain any of the parameters found within the CPF, as described in this book. You can include comments using the # (pound sign) symbol.

A merge file contains only the parameters you wish to change; it is not necessary to include those you are not changing. Unlike the CPF, a merge file can contain duplicates of the same section and parameter; InterSystems IRIS uses whichever value appears closer to the end of the file to update the CPF. This allows you to use a template merge file with generally desired values at the beginning, adding settings you want to override at the end to customize the template for particular instances or groups of instance.

As an example, you can create a merge file called config\_merge.cpf that contains the desired configuration values. Most of the default values are sufficient for your deployment, but the default [generic memory heap](#) size of 37,568 KB is too low, and you want to specify the size of the [database cache](#). The config\_merge.cpf file would contain only these lines:

```
[config]
globals=0,0,800,0,0,0
gmheap=256000
```

When it is time to deploy an instance, set the value of **ISC\_CPF\_MERGE\_FILE** to the full path of config\_merge.cpf. For a silent installation using an installation kit, this would look like:

```
sudo ISC_PACKAGE_INSTANCENAME="IRIS" ISC_PACKAGE_INSTALLDIR="/intersystems/iris/"
ISC_CPF_MERGE_FILE=/mergefiles/custom-params.config ./irisinstall_silent
```

Before the first time InterSystems IRIS runs any start procedures, it merges the changes from config\_merge.cpf into iris.cpf. Because the merge occurs prior to the first startup, no restart is necessary for the changes to take effect.

For information about using a merge file with ICM, see the [Deploying with Customized InterSystems IRIS Configurations](#) section of the “ICM Reference” chapter in *InterSystems Cloud Manager Guide*.

## 1.3 Configuration Security

If InterSystems IRIS startup detects that the configuration parameter file has been modified by a text editor since the last time InterSystems IRIS was started, and if [Configuration Security is enabled](#), InterSystems IRIS startup requests a username and password to validate the changes. The username supplied must have **%Admin\_Manage:Use** privileges. If the appropriate username and password cannot be provided, InterSystems IRIS allows the operator to choose as follows:

1. Re-enter the username and password.
2. Start using the last known good configuration.
3. Abort startup.

If the operator chooses option 2, InterSystems IRIS renames the parameter file that was invoked at startup (*file.cpf*) with the suffix \_rejected (*file.cpf\_rejected*). InterSystems IRIS then overwrites the *file.cpf* with the last known good configuration (\_LastGood\_.cpf) and starts up using this configuration.

For more information on other system-wide security parameters, see the “[System Management and Security](#)” chapter in the *Security Administration Guide*.

## 1.4 Parameter Descriptions

Each parameter reference page in this book includes most of the following sections:

- *Synopsis* – The CPF section that contains this parameter, followed by a summary of its syntax. Beneath this, a description of valid inputs and the default value.
- *Description* – A formal description of the parameter. May include examples of valid inputs or guidelines for choosing values.
- *Changing this Parameter* – The various ways to change this parameter, either programmatically or using the browser-based Management Portal.
- *See Also* – Links to related parameters and relevant documentation.

# [ComPorts]

This topic describes the parameters found in the [ComPorts] section of the CPF. The [ComPorts] section applies to Windows systems only.

# COMn

Define default settings for COM ports. Windows systems only.

```
[ComPorts]      COMn=a ; b
```

## Description

The [ComPorts] section contains an entry for each COM port. These entries define the default settings for COM ports, and enable remote logins to InterSystems IRIS® data platform through locally connected or modem connected serial ports. The number *n* refers to the physical COM port number. If the [ComPorts] section contains enough entries, *n* may have multiple digits.

The parameters within COMn apply to Windows platforms only.

Each COMn entry provides two semicolon-separated values that define the default settings for COM port number *n*. These are:

- *a* — A set of COM port control parameters (data bits, parity, etc.) in byte-positional format. Byte position is one-based. From left to right:

Byte 1: Modem Control	<ul style="list-style-type: none"> <li>- '1' Use modem control (selected in the 1801X11 example below)</li> <li>- '0' Do not use modem control</li> <li>- ' ' No change to modem control (this is the default)</li> </ul>
Byte 2: Data Bits	<ul style="list-style-type: none"> <li>- '5' 5 data bits</li> <li>- '6' 6 data bits</li> <li>- '7' 7 data bits</li> <li>- '8' 8 data bits (selected in the 1801X11 example below)</li> <li>- ' ' No change to bit size (this is the default)</li> </ul>
Byte 3: Parity	<ul style="list-style-type: none"> <li>- '0' No parity (selected in the 1801X11 example below)</li> <li>- '1' Odd parity</li> <li>- '2' Even parity</li> <li>- '3' Mark parity</li> <li>- '4' Space parity</li> <li>- ' ' No change to the parity setting (this is the default)</li> </ul>
Byte 4: Stop Bits	<ul style="list-style-type: none"> <li>- '1' 1 stop bit (selected in the 1801X11 example below)</li> <li>- '5' 1.5 stop bits</li> <li>- '2' 2 stop bits</li> <li>- ' ' No change to the stop bit setting (this is the default)</li> </ul>
Byte 5: Flow Control	<ul style="list-style-type: none"> <li>- 'x' Use Xon/Xoff flow control (selected in the 1801X11 example below)</li> <li>- 'c' Use CTS/RTS flow control</li> <li>- 'd' Use DSR/DTR flow control</li> <li>- ' ' No change to flow control (this is the default)</li> </ul>
Byte 6: DTR Setting	<ul style="list-style-type: none"> <li>- '0' Disable DTR (set it off, keep it off)</li> <li>- '1' Enable DTR (set it on, keep it on) (selected in the 1801X11 example below)</li> <li>- ' ' No change to the DTR state (this is the default)</li> </ul>

## Byte 7: \$ZA Error Reporting

- '0' Disable \$ZA error reporting
- '1' Enable \$ZA error reporting (selected in the 1801X11 example below)
- ' ' No change to \$ZA error reporting (this is the default)

- $b$  — The baud rate. If not supplied, the default baud rate is 19200.

**Examples**

The following example of a [ComPorts] section shows how spaces can be used as values within the COM port control parameter:

```
[ComPorts]
COM1=      ;19200
```

The first example uses all defaults for the COM port control parameters, by providing seven spaces before the semicolon separator. The meaning is: No change to modem control, no change to bit size, no change to parity setting, no change to stop bit setting, no change to Flow control, no change to DTR state, no change to \$ZA error reporting.

```
[ComPorts]
COM2=1801X11;19200
```

The second example provides a value of 1801X11 for the COM port control parameters. The meaning is: Use modem control, 8 data bits, no parity, 1 stop bit, use Xon/Xoff flow control, enable DTR, enable \$ZA error reporting.

**Changing This Parameter**

You can change this parameter by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# [config]

This topic describes the parameters found in the [config] section of the CPF.

## LibPath

Set the LD\_LIBRARY\_PATH environment variable. UNIX® systems only.

```
[config]    LibPath=path
```

*path* is a path to a valid directory. By default, no directory is listed.

### Description

LibPath is used for UNIX® systems only. This parameter sets the LD\_LIBRARY\_PATH environment variable (DYLD\_LIBRARY\_PATH on macOS) to *path* to search for third-party shared libraries. If you modify this setting, you must restart the instance for the change to take effect.

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

### Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration > Configuration > Additional Settings > Advanced Memory**), in the **LibPath** row, select **Edit**. Enter a path.

Instead of using the Management Portal, you can change LibPath in the Config.config class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# MaxServerConn

Set the maximum number of connections from ECP clients.

```
[config]    MaxServerConn=n
```

*n* is an integer in the range 0—254. The default value is 1.

## Description

MaxServerConn is the 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. If you modify this setting, you must restart the instance for the change to take effect.

## Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration > Configuration > Additional Settings > Advanced Memory**), in the **MaxServerConn** row, select **Edit**. Enter a value.

You can also change this parameter on the **ECP Settings** page of the Management Portal (**System Administration > Configuration > Connectivity > ECP Settings**). In the **This System as an ECP Data Server** column, edit **Maximum number of application servers**.

Instead of using the Management Portal, you can change MaxServerConn in the Config.config class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# MaxServers

Set the maximum number of connections to ECP servers.

```
[config]    MaxServers=n
```

*n* is an integer in the range 0—254. The default value is 2.

## Description

**MaxServers** is the 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. If you modify this setting, you must restart the instance for the change to take effect.

## Changing This Parameter

On the page **System Administration > Configuration > Connectivity > ECP Settings**, in the **This System as an ECP Application Server** column, edit **Maximum number of data servers**.

Instead of using the Management Portal, you can change **MaxServers** in the Config.config class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# Path

Add directories to the default path environment variable. UNIX® systems only.

```
[config]    Path=directory1[:directory2]
```

*directory* is the full path to a valid directory. By default, no directories are listed.

## Description

On UNIX® systems, InterSystems IRIS® data platform processes started during InterSystems IRIS system startup are assigned a UNIX® PATH environment variable set by default to:

```
PATH=/usr/bin:/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin
```

Processes that use this path include InterSystems IRIS system daemons, processes started by the SYSTEM^%ZSTART routine, and processes which will be started by the Super Server (such as JDBC/ODBC servers).

Customer applications may require that the PATH environment variable for these processes have additional search directories appended to the default PATH provided by InterSystems IRIS. You can append directories to this path using the Path variable. If you modify this setting, you must restart the instance for the change to take effect.

**Note:** Terminal processes do not set their PATH this way; their PATH should be set by their login scripts.

## Example

```
Path=/usr/customerapp/bin
```

This sets the PATH environment variable for system processes to

```
PATH=/usr/bin:/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:/usr/customerapp/bin
```

## Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration > Configuration > Additional Settings > Advanced Memory**), in the **Path** row, select **Edit**. Enter a directory name.

Instead of using the Management Portal, you can change Path in the Config.config class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## bbsiz

Set the maximum memory per process.

```
[config]    bbsiz=n
```

*n* is an integer in the range 128—2,147,483,647 (KB). The default value is 262,144.

### Description

`bbsiz` is the maximum memory allocation permitted for a process (in kilobytes). This amount of process private memory is used for symbol table allocation and various other memory requirements, including I/O devices. It is allocated in increasing extents as required by the application until the maximum is reached. When a process starts, its initial allocation is 128 kilobytes.

Once this memory is allocated to the process, it is generally not deallocated until the process exits. However, when a large amount of memory is used (for example greater than 32MB) and then freed, InterSystems IRIS® data platform attempts to release deallocated memory back to the operating system where possible.

### Changing This Parameter

On the **Memory and Startup** page of the Management Portal (**System Administration > Configuration > System Configuration > Memory and Startup**), enter a number of kilobytes in the **Maximum Per-Process Memory (KB)** row.

Instead of using the Management Portal, you can change `bbsiz` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# console

Set the path to the messages log file (messages.log).

```
[config]    console=VMSConsoleTerminal,ConsoleFile
```

*ConsoleFile* is the full path to a valid directory. By default, no directory is listed.

## Description

console contains two comma-separated values that configure the messages.log file, as described below:

### **VMSConsoleTerminal**

Not in use.

### **ConsoleFile**

The path to the messages.log file, where InterSystems IRIS® data platform logs system messages. If no value is specified, InterSystems IRIS writes to *install-dir/mgr/messages.log*.

ConsoleFile is the second value of the [console](#) parameter.

You can view the messages log on **Messages Log** page of the Management Portal ([System Operation > System Logs > Messages Log](#)). To further configure the messages.log file, see [MaxConsoleLogSize](#) parameter.

## Changing This Parameter

On the **Advanced Memory** page of the Management Portal ([System Administration > Configuration > Additional Settings > Advanced Memory](#)), in the **ConsoleFile** row, select **Edit**. Enter a directory path.

Instead of using the Management Portal, you can change ConsoleFile in the Config.config class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## errlog

Set the maximum number of entries in the error log.

```
[config]    errlog=n
```

*n* is an integer in the range 10—10,000. The default value is 500.

### Description

`errlog` is the maximum number of entries in the InterSystems IRIS® data platform system error log (see [InterSystems IRIS System Error Log](#) in the “Monitoring InterSystems IRIS Using the Management Portal” chapter in *Monitoring Guide* for more information). The log file expires old entries as this limit is reached.

### Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration > Configuration > Additional Settings > Advanced Memory**), in the `errlog` row, select **Edit**. Enter a number of entries.

Instead of using the Management Portal, you can change `errlog` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# globals

---

Allocate shared memory to the database cache.

```
[config]    globals=0,0,c,d,e,f
```

## Description

`globals` contains six comma-separated values that specify how much shared memory to allocate to the database cache for each block size. From left to right, each value is the number of megabytes allocated for:

- *a* – no longer used, always 0
- *b* – no longer used, always 0
- *c* – 8-kilobyte blocks
- *d* – 16-kilobyte blocks
- *e* – 32-kilobyte blocks
- *f* – 64-kilobyte blocks

When all six values are set to 0, the system automatically allocates space based on an estimate of the machine capability and shared memory resources. On a 64-bit system, there is a limit of 16 TB.

For more information, see [Allocating Memory to the Routine and Database Caches](#) in the “Configuring InterSystems IRIS” chapter in *System Administration Guide*.

**Note:**

## Changing This Parameter

To create a database with a block size other than the default 8-KB blocks, you must enable additional block sizes using the [DBSizesAllowed](#) parameter in the [Startup] section.

On the **Memory and Startup** page of the Management Portal ([System Administration > Configuration > System Configuration > Memory and Startup](#)), select the option to **Configure Memory Settings Manually**. In the **Memory Allocated for [block size]** **Database Cache (MB)** field, enter the number of megabytes to allocate for the specified block size.

Instead of using the Management Portal, you can change `globals` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

- [DBSizesAllowed](#) parameter in the [Startup] section of this reference.
- [Allocating Memory to the Routine and Database Caches](#) in the “Configuring InterSystems IRIS” chapter in *System Administration Guide*.
- [Large Block Size Considerations](#) in the “Configuring InterSystems IRIS” chapter in *System Administration Guide*.

## gmheap

Set the size of the generic memory heap.

```
[config] gmheap=n
```

*n* is an integer in the range 2048—1,073,741,760 (KB). The default is 37,568.

### Description

gmheap is the size in kilobytes of the generic memory heap (also known as the shared memory heap or SMH) for InterSystems IRIS® data platform. 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 (**System Operation > System Usage**, then click the **Shared Memory Heap Usage** button); see [Generic \(Shared\) Memory Heap Usage](#) in the “Monitoring InterSystems IRIS Using the Management Portal” chapter in *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®). (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:

- 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 in *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 in *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.

### Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration > Configuration > Additional Settings > Advanced Memory**), in the **gmheap** row, select **Edit**. Enter a number of kilobytes.

Instead of using the Management Portal, you can change gmheap in the Config.config class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

If you edit the gmheap setting, you must restart InterSystems IRIS to apply the change.

# history

Define command line recall options.

```
[config]    history=LineRecallEntries,LineRecallBuffer
```

*LineRecallEntries* is an integer in the range 0—256. The default value is 32.

*LineRecallBuffer* is an integer in the range 0—8192 (bytes). The default is 1024.

## Description

`history` contains two comma-separated values which configure command line recall features: **LineRecallEntries** and **LineRecallBuffer**.

### **LineRecallEntries**

Maximum number of entries held in the command line/read line recall buffer, subject to the space limitation in the second parameter. The range is 0–256 entries. The default is 32 entries.

`LineRecallEntries` is the first value in the [history](#) parameter.

### **LineRecallBuffer**

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

`LineRecallBuffer` is the second value in the [history](#) parameter.

## Changing This Parameter

On the **Advanced Memory** page of the Management Portal ([System Administration > Configuration > Additional Settings > Advanced Memory](#)), edit the values for **LineRecallEntries** or **LineRecallBuffer**.

Instead of using the Management Portal, you can change `history` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

If you edit these setting, you must restart InterSystems IRIS® data platform to apply the change.

## ijcbuf

Set the size of an InterJob Communication buffer.

```
[config]    ijcbuff=n
```

*n* is an integer in the range 512—8192 (bytes). InterSystems recommends you use the default of 512.

### Description

`ijcbuf` is the number of bytes allocated for each InterJob Communication (IJC) buffer. For details, see the “[Interprocess Communication](#)” chapter in the *I/O Device Guide*. Also see the parameter [ijcnum](#).

### Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration > Configuration > Additional Settings > Advanced Memory**), in the **ijcbuf** row, select **Edit**. Enter a number of bytes.

Instead of using the Management Portal, you can change `ijcbuf` in the Config.config class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

If you edit this setting, you must restart InterSystems IRIS® data platform to apply the change.

# ijcnum

Set the number of InterJob Communication devices.

```
[config]    ijcnum=n
```

*n* is an integer in the range 0—256. InterSystems recommends you use the default of 16 devices.

## Description

`ijcnum` is the number of InterJob Communication (IJC) devices. Each device corresponds to one InterJob Communication buffer of the size defined by `i jcbuff`. For details, see the “[Interprocess Communication](#)” chapter in the *I/O Device Guide*.

## Changing This Parameter

On the **Advanced Memory** page of the Management Portal ([System Administration > Configuration > Additional Settings > Advanced Memory](#)), in the `ijcnum` row, select **Edit**. Enter a number of devices.

Instead of using the Management Portal, you can change `ijcnum` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

If you edit this setting, you must restart InterSystems IRIS® data platform to apply the change.

## jrnbufs

Allocate memory for journal buffers.

```
[config]    jrnbufs=n
```

*n* is an integer in the range 8—1024 (MB). The default value is 64.

### Description

`jrnbufs` is the amount of memory allocated for journal buffers. Increasing this setting means increasing the amount of journal data that can be held in memory, which improves journaling performance but increases the amount of journal data that could potentially be lost in the event of a system failure.

### Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration > Configuration > Additional Settings > Advanced Memory**), in the `jrnbufs` row, select **Edit**. Enter a number of megabytes.

Instead of using the Management Portal, you can change `jrnbufs` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

If you edit this setting, you must restart InterSystems IRIS® data platform to apply the change.

# locksiz

Set the maximum size of shared memory for locks.

```
[config]    locksiz=n
```

*n* is an integer with a minimum value of 65,536 (bytes). The default value is 16,777,216.

## Description

`locksiz` is the size of memory in bytes allocated for locks. The system rounds up the value to the next multiple of 64 kilobytes. The default is 16,777,216 bytes. (On the IBM AIX and HP-UX platforms, the default is 33,554,432 bytes.)

Since memory used to allocate locks is taken from [gmheap](#) (the Generic Memory Heap), you cannot use more memory for locks than exists in [gmheap](#). If you need more room for the lock table, increase the [gmheap](#) parameter as well. If you edit this setting, changes take effect immediately.

## Changing This Parameter

On the **Advanced Memory** page of the Management Portal ([System Administration > Configuration > Additional Settings > Advanced Memory](#)), in the **locksiz** row, select **Edit**. Enter a number of bytes.

Instead of using the Management Portal, you can change `locksiz` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# memlock

Specify locking shared memory or the text segment into memory.

```
[config]    memlock=n
```

*n* is a set of bit flags. By default, the *n* is 0 (all flags are set to false).

## Description

memlock is a set of bit flags that controls how InterSystems IRIS® data platform allocates shared memory. At startup, InterSystems IRIS allows a shared memory segment for use by control structures, global buffers, routine buffers, and generic memory heap. The memlock parameter offers detailed control over how that allocation occurs.

By default (*n* = 0), InterSystems IRIS attempts to allocated shared memory from large pages on platforms that support large pages (Windows, Linux, and AIX) as follows:

1. Request large pages, if allowed. Large pages are automatically locked in physical memory at the operating system level.
2. If unable to allocate the full amount of configured memory in large pages, request standard (small) pages. InterSystems IRIS does not attempt to lock standard pages into physical memory.
3. If unable to allocate the full amount of configured memory in small pages, reduce the allocation by one eighth (1/8) and begin again with step 1.

The following bit flags modify this process as described below:

### 1 (LockSharedMemory)

This `memlock` flag indicates whether shared memory is locked in physical memory when large pages are not being used. By default, it is not. This applies to all operating systems except for Microsoft Windows and macOS.

### 8 (LockTextSegment)

This `memlock` flag indicates whether the text segment (the InterSystems IRIS executable code space) is locked in physical memory (on some UNIX platforms). By default, it is not.

### 32 (LargePagesDisabled)

This `memlock` flag indicates whether to disable large/huge pages for shared memory on platforms that support them. By default, large/huge pages are used.

When this flag is off on platforms supporting large pages, InterSystems IRIS attempts to allocate memory in large pages and switches back to standard pages if large pages cannot be allocated at the requested size. Technically, this means that InterSystems IRIS adopts a neutral disposition towards page size, taking no action to request large pages.

### 64 (LargePagesRequired)

This `memlock` flag indicates whether to require use of large/huge pages for shared memory on platforms that support them (Windows, AIX, and Linux). By default, it is not required. This flag is ignored on other platforms, or if large pages are disabled by the `LargePageDisabled` flag.

When `LargePagesRequired` is True (and not ignored), if memory cannot be allocated in large/huge pages, startup is aborted rather than falling back to small pages. InterSystems IRIS retries with a small reduction in memory size, but not as much of a reduction as could occur in absence of this flag.

## 128 (*BackoffDisabled*)

This [memlock](#) flag indicates, on failure to allocate memory, whether to retry with a reduced amount. By default, it does retry. If this flag is True and memory cannot be allocated at its configured size, startup is aborted.

### Changing This Parameter

On the **Advanced Memory** page of the Management Portal ([System Administration > Configuration > Additional Settings > Advanced Memory](#)), choose true or false for **BackoffDisabled**, **LargePagesDisabled**, **LargePagesRequired**, **LockSharedMemory**, and **LockTextSegment**.

Instead of using the Management Portal, you can change `memlock` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

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

## netjob

Allow remote job requests.

```
[config]    netjob=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

### Description

When `net job` is enabled (*n* = 1), incoming remote job requests via ECP are honored on this server.

### Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration > Configuration > Additional Settings > Advanced Memory**), in the **netjob** row, select **Edit**. Choose true or false.

Instead of using the Management Portal, you can change `net job` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# nlstab

Set the number of NLS collation tables.

```
[config]    nlstab=n
```

*n* is an integer in the range 0—64. The default value is 50.

## Description

**nlstab** is the number of NLS collation tables for which to allocate when InterSystems IRIS® data platform starts up. This parameter refers to loadable national collation tables and does not include built-in collations.

## Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration > Configuration > Additional Settings > Advanced Memory**), in the **nlstab** row, select **Edit**. Enter a number of tables.

Instead of using the Management Portal, you can change **nlstab** in the **Config.config** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# overview

Review InterSystems IRIS® data platform platform and version summary.

```
[config] overview=a~b
```

*a* and *b* are read-only strings.

## Description

`overview` displays two tilde-separated (~) values which provide platform and version information. From left to right:

- *a*: The specific operating system on which InterSystems IRIS is running.
- *b*: The general type of the operating system.

## Examples

```
overview=Windows (Intel)~Windows
```

```
overview=Linux (Intel)~UNIX®
```

---

## pijdir

---

Not in use.

### Description

Not in use.

## routines

Allocate shared memory for caching routine buffers.

```
[config]    routines=n
[config]    routines=n1, n2, n3, n4, n5, n6
```

### Description

`routines` specifies the amount of memory to allocate for caching routine buffers. You can do this in two ways — either specifying a single value or specifying values for all six buffer sizes.

You can specify only a single value, *n*, for the total size, in megabytes, of shared memory that InterSystems IRIS® data platform allocates for routine buffers. InterSystems IRIS divides this memory into the various buffer sizes for optimum performance. While the default value is 0, InterSystems IRIS always allocates a minimum of 35 MB. The maximum value depends on how much shared memory is allocated for global buffers and other uses. The total shared memory size cannot exceed 1023 MB.

You can instead specify the amount of memory to allocate for each routine buffer size. If you use this format, you must specify all six values; if fewer than six are given, InterSystems IRIS reverts to the first format, using *n1* as the value for *n*. The six values represent the following:

- *n1* is the number of MB for 2 KB routine buffers
- *n2* is the number of MB for 4 KB routine buffers
- *n3* is the number of MB for 8 KB routine buffers
- *n4* is the number of MB for 16 KB routine buffers
- *n5* is the number of MB for 32 KB routine buffers
- *n6* is the number of MB for 64 KB routine buffers

The total number of buffers cannot exceed 65,528. If the number of buffers allocated exceeds this number, it is reduced to the maximum. Some of the values can be 0, but the next minimum value is 430 (if a smaller number is specified, the system adjusts up to 430).

See [Memory and Startup Settings](#) in the “Configuring InterSystems IRIS” chapter in *System Administration Guide* for more information

### Examples

```
routines=256
```

Allocate 256 MB for routine buffers, divided into various sizes by an internal formula.

```
routines=2000,2000,0,500,1000,1000
```

Allocate 2,000 MB each for 2 KB and 4 KB buffers, 500 MB for 16 KB buffers, and 1,000 MB each for 32 KB and 64 KB buffers

```
routines=0
```

To allocate a minimum number of buffers, enter a single value of 0.

### Changing This Parameter

On the **Memory and Startup** page of the Management Portal (**System Administration > Configuration > System Configuration > Memory and Startup**), select the option to **Configure Memory Settings Manually**. In the **Memory Allocated for Routine Cache (MB)** field, enter a number of megabytes.

Instead of using the Management Portal, you can change `routines` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## udevtabsiz

Set the maximum size of the device table.

```
[config] udevtabsiz=n
```

*n* is an integer in the range 0—65,535 (bytes). The default value is 24,576.

### Description

`udevtabsiz` is the maximum size in bytes of the device table. This table maps device numbers (traditional logical unit numbers) to device names, so that ObjectScript code can open devices by number.

### Changing This Parameter

On the page [System Administration > Configuration > Additional Settings > Advanced Memory](#), in the `udevtabsiz` row, select **Edit**. Enter a number of bytes.

Instead of using the Management Portal, you can change `udevtabsiz` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# wijdir

Set the Write Image Journal files directory.

```
[config]    wijdir=n
```

*n* is the full path to a valid directory. By default, no directory is listed.

## Description

`wijdir` is the name of the directory where the write image journal file is stored. The directory name may be up to 214 characters long. InterSystems recommends that the journal directory be located in a different partition from your databases.

If no value is specified, InterSystems IRIS® data platform uses the *install-dir/mgr* directory.

## Changing This Parameter

On the **Journal Settings** page of the Management Portal (**System Administration > Configuration > System Configuration > Journal Settings**), in the **Write image journal directory** row, select **Browse**. Select a directory name.

Instead of using the Management Portal, you can change `wijdir` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## targwijsz

Set the desired size of the wij.

```
[config] targwijsz=n
```

*n* is an integer. The default is 0.

### Description

When **targwijsz** is set, the WIJ expands to the specified size (in MB). When set to 0, the WIJ grows as needed, based on activity. The value is an integer, but fractional input is accepted and silently truncated (for example, 35.5 becomes 35).

For more information, see the [Write Image Journaling and Recovery](#) chapter in *Data Integrity Guide*.

**Note:** Setting this target size ensures that disk space is allocated for the WIJ early in the start-up process. If sufficient space is not allocated early and there is not enough available space for the WIJ, the system may encounter problems. Allocating space for WIJ is an advanced configuration setting. If you encounter issues with this, contact the InterSystems [Worldwide Resource Center](#).

### Changing This Parameter

On the **Advanced Memory Settings** page of the Management Portal (**System Administration > Configuration > Additional Settings > Advanced Memory**), click the **Edit** link in the **targwijsz** row. The **Edit: targwijsz** page provides details about the setting and allows you to change its value.

Instead of using the Management Portal, you can change **targwijsz** in the **Config.config** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# zfheap

---

Define the size and configuration of the \$ZF heap.

```
[config]    zfheap=ZFString,ZFSize
```

*ZFString* is an integer in the range 0—32,767. The default is 0.

*ZFSize* is an integer in the range 0—270,336 9 (bytes). The default is 0.

## Description

`zfheap` contains two comma-separated values which configure the \$ZF heap: **ZFString** and **ZFSize**. For more details about the \$ZF heap, see the chapter “[Creating a Callout Library](#)” in *Using the Callout Gateway*.

### ZFString

This `zfheap` parameter is the number of characters InterSystems IRIS® data platform allows for a single string parameter on the \$ZF heap. How many bytes this actually requires depends on whether you are using Unicode (2-byte characters) or 4-byte characters on UNIX®. When set to 0, InterSystems IRIS automatically uses the system default value for `ZFString`, which is 32,767.

### ZFSize

This `zfheap` parameter is the number of bytes InterSystems IRIS® data platform 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. When set to 0, InterSystems IRIS automatically calculates an appropriate value for `ZFSize`, based on the value of `ZFString`. The formula is as follows:

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

### Examples

Using the default value for `ZFString`, which is 32,767:

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

## Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration > Configuration > Additional Settings > Advanced Memory**), edit the values for `ZFString` or `ZFSize`. InterSystems recommends that this parameter be set to 0 , 0.

Instead of using the Management Portal, you can change `zfheap` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

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



# [ConfigFile]

This topic describes the parameters found in the [ConfigFile] section of the CPF.

# Version

Review the version of the configuration parameter file.

```
[ConfigFile] Version=n
```

*n* is a string of numbers, usually in dot format.

## Description

Version is the configuration parameter (CPF) file version number. This value is independent of the product release number. It is updated automatically during product upgrade or installation. The number refers to the syntax and structure of the parameter file, and not to the changes that you might make to the parameter values within the file.

**Important:** Changing this value from the default can cause InterSystems IRIS® data platform to fail on startup. To differentiate between multiple user-edited CPFs, see the [Comments](#) section in the “Introduction to the Configuration Parameter File” chapter in this book.

# [Databases]

This topic describes the parameters found in the [Databases] section of the CPF.

# Database

Define InterSystems IRIS® data platform databases.

```
[Databases]      Name=a,b,c,d,e, f
```

*Name* is a string. *a*, *b*, *c*, *d*, *e*, and *f* are as described below.

## Description

The [Databases] section of the configuration parameter (CPF) file contains an entry for every database connected to the instance. Each entry has up to six comma-separated arguments that define a database. The only argument that is required is the first, the directory specification. If the other arguments are not specified, then the default is used. The values are:

- *a* — Full directory specification for database file. Required.
- *b* — Remote system name (empty for local system). If the system is remote (nonempty name), subsequent fields are ignored. Default is the local system.
- *c* — Must mount at startup (1 or 0). When enabled, the database must be mounted at startup, or startup fails. Default is 0 (database is not mounted at startup).
- *d* — This value is ignored and may be left blank.
- *e* — This value is ignored and may be left blank.
- *f* — Stream Location. Directory where the streams associated with this database go. The default is blank — "" (which Intersystems recommends.) This default location is in the subdirectory stream, underneath the database directory. For example, for a database located in c:\abc, the default stream location is c:\abc\stream.

By default, even if they are not specified, the IRISSYS, IRISLIB, IRISTEMP, and IRISAUDIT databases must be local to the system and mount at startup.

## Examples

In the [Databases] section, each line is a database with arguments *Name=a,b,c,d,e,f*. Here is an example from Windows:

```
[Databases]
IRISSYS=c:\InterSystems\IRIS\mgr\
IRISLIB=c:\InterSystems\IRIS\mgr\irislib\
IRISTEMP=c:\InterSystems\IRIS\mgr\iristemp\
IRISAUDIT=c:\InterSystems\IRIS\mgr\irisaudit\
USER=c:\InterSystems\IRIS\mgr\user\
SALES=c:\sales\,SALESEVER
; Database is on system SALESEVER
BILLING=/usr/billing/,,1
; Database is local and mount required
```

Example of remote databases without mirroring:

```
PRDAUDIT=c:\InterSystems\IRIS\mgr\prdaudit\,,PRD
PRDDCIFC=\InterSystems\IRIS\mgr\prddata\,,PRD
PRDERR=\InterSystems\IRIS\mgr\prderr\,,1
```

Example of remote databases with mirroring. Note in this mirrored example the two remote databases, both mirrored and nonmirrored, are formatted differently than they would be in a nonmirrored environment.

```
PRDAUDIT=:mirror:PRDMIRROR:PRDAUDIT,PRD - Mirrored remote database
PRDDCIFC=:ds:PRDCIFC,PRD - Nonmirrored remote database
PRDERR=\InterSystems\IRIS\mgr\prderr\,,1 - Local database
```

## Changing This Parameter

On the **Local Databases** page of the Management Portal (**System Administration > Configuration > System Configuration > Local Databases**), to add a new entry, select **Create New Database**. To edit an existing entry, select **Edit** in that entry's row.



# [Debug]

The configuration parameter file may include a [Debug] section. The [Debug] settings can be used for different kind of diagnostics. This topic describes two parameters that may be found in the [Debug] section of the configuration parameter file. All other [Debug] settings are InterSystems proprietary and this book does not document them.

# Dumpstyle

Specify the style of core dump.

```
[Debug]      dumpstyle=n
```

*n* is an integer in the range 0—4. The default value is 3.

## Description

When InterSystems IRIS® data platform performs a core dump, you can set the style of the dump using this option. Values and their meanings are listed in the following.

**Note:** On Unix®, all dump styles generate a core file. The process cleans itself up as much as possible before exiting.

- 0
  - On Windows this is the *pid.dmp* file.
- 1
  - On Windows this is a Windows minidump file (type = `MiniDumpWithFullMemory`) named `cachefpid.dmp`, which can be read by **WinDbg**, a Microsoft debugger. This is the most complete dump option, but it can create a huge dump file.
- 2
  - On Windows this is the old style exception processing where the process does minimal cleanup (`deqallresources` and `GRETRLEASE`) and then resignals the exception. The intention here is to catch the exception in a debugger and preserve as much information as possible to analyze.
  - On Unix® this detaches shared memory before the Abort, so the core file does not contain the shared memory area.
- 3
  - On Windows this is a Windows minidump file (type = `MiniDumpWithDataSegs | MiniDumpWithPrivateReadWriteMemory | MiniDumpWithIndirectlyReferencedMemory`) named `cacheipid.dmp`, which can be read by **WinDbg**. This creates a fairly large but useful dump file. This is the new default on Windows if `dumpstyle` is not specified.
- 4
  - On Windows this is a Windows minidump file (type = `MiniDumpNormal`) named `cachempid.dmp`, which can be read by **WinDbg**. This creates a small dump file containing minimal information.

The active value can be changed with `$system.Config.ModifyDumpStyle(NewValue)`. This changes the value for all new InterSystems IRIS processes. It does not change the value in `iris.cpf`.

## Changing This Parameter

You can change `Dumpstyle` by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# Semsperset

Set number of semaphores allocated per set.

```
[Debug]      semsperset=n
```

*n* is an integer. The default value is 0.

## Description

**Semsperset** is the number of semaphore InterSystems IRIS® data platform should allocate per set. When this parameter is set to 0, InterSystems IRIS allocates the minimum number of semaphore sets by maximizing the number of semaphores per set (see [Semaphores in InterSystems Products](#)). The **semsperset** parameter in the CPF can be used to specify a lower number of semaphores per set.

## Changing This Parameter

You can change **Semsperset** by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).



# [Devices]

This topic describes the parameters found in the [Devices] section of the CPF.

# Devices

Define device types.

```
[Devices]      Name=a^b^c^d^e^f^g^h
```

## Description

The [Devices] section of the configuration parameter (CPF) file contains an entry for every device detected by InterSystems IRIS® data platform. Each entry has a *Name*, which is the defined device title or number, and eight strings separated by up-arrows (^) that define a device. The maximum length of all strings is 128 characters, except for the *Description* (g) string, which can be up to 256 characters. The entries are as follows

- (a) *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

- (b) *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.
- (c) *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.
- (d) *Prompt*— Choose a prompt: valid inputs are 1, 2, or NULL (a blank or empty field). 1 corresponds to **Auto-use this device if it is the current device**. 2 corresponds to **Auto-use this device with predefined settings** (predefined Right Margin and Parameter settings). NULL corresponds to **Show device prompt** (the user sees the device selection prompt with the default device defined).
- (e) *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 [ObjectScript Reference](#).

- (f) *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 InterSystems IRIS 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 in *I/O Device Guide*. The topic of most interest is “%IS Mnemonics,” which describes the conventions for entering the “A” code for %IS.

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

## Examples

In the [Devices] section, each entry *Name=a^b^c^d^e^f^g^h* appears all on one line:

```
[Devices]
0=0^TRM^C-Terminal^^^^Principal device^
2=2^SPL^PK-DEC^^^^Spool LA120^
SPOOL=2^SPL^PK-DEC^^^^Spool LA120^
TERM=0^TRM^C-Terminal^^^^Windows Console^
| PRN |=| PRN|^OTH^P-DEC^^"W"^^Windows Printer^
| TNT |=0^TRM^C-VT220^^^^Principal device^
| TRM |=0^TRM^C-Terminal^^^^Windows Console^
```

## Changing This Parameter

On the **Devices** page of the Management Portal ([System Administration > Configuration > Device Settings > Devices](#)) is a list of existing devices. Select **Create New Device**, **Edit**, or **Delete** to modify the list.



# [DeviceSubTypes]

This topic describes the parameters found in the [DeviceSubTypes] section of the configuration parameter file.

# DeviceSubTypes

Define device subtypes.

```
[DeviceSubTypes]      Name=n=a^b^c^d^e^f^g^h^i
```

## Description

The [DeviceSubTypes] section of the configuration parameter (CPF) file contains an entry for each subtype configured for this installation. Each entry has a *Name*, as well as nine values separated by up-arrows (^) that define a device subtype. The entries are as follows

- (a) *RightMargin*—The number that represents the location of the right margin. Device output wraps at that number of characters.
- (b) *FormFeed*—The ASCII code that represents a form feed on the selected device in the form #,\$C(code1,code2...). This setting is used by the InterSystems IRIS® data platform CHUI utilities.
- (c) *ScreenLength*—The number of lines that comprise one screen or page for the device.
- (d) *Backspace*—The ASCII code that represents the backspace character on the selected device in the form \$C(code1). This setting is used by the InterSystems IRIS CHUI utilities.
- (e) *CursorControl*—The ASCII code that represents the cursor on the selected device in the form \$C(code1).
- (f) *EraseEOL*—The ASCII code that represents erasing the end of line characters on this device in the form \$C(code1,code2).
- (g) *EraseEOF*—The ASCII code that represents erasing the end of file character on the selected device in the form \$C(code1,code2...).
- (h) *ZU22FormFeed*—The ASCII code that represents a form feed on the selected device in the form \$C(code1,code2). This setting is used by InterSystems IRIS for Terminal output.
- (i) *ZU22Backspace*—The ASCII code that represents a backspace on the selected device in the form \$C(code1). This setting is used by InterSystems IRIS for Terminal output.

Default values depend on the device type.

## Examples

The following is a sample [DeviceSubTypes] section. This example wraps long lines to fit them onto the viewing page. In the .cpf file itself, each entry appears all on one line:

```
[DeviceSubTypes]
C-ANSI=80^#, $C(27,91,72,27,91,74)^25^$C(8)^W $C(27,91)_-(DY+1)_;"_-(DX+1)_-"H"
S $X=DY^$C(27,91,74)^$C(27,91,75)^$C(27,91,72,27,91,74)^$C(8,32,8)
C-Terminal=80^#, $C(27,91,72,27,91,74)^24^$C(8)^W $C(27,91)_-(DX+1)_;"_-
(DX+1)_-"H" S $X=DX, $Y=DY^$C(27,91,74)^$C(27,91,75)^$C(27,91,72,27,91,74)^
$C(8,32,8)
C-TV925=80^#, $C(27,44)^24^$C(8)^W $C(27,61,DY+32,DX+32) S $X=DX, $Y=DY^^^
$C(27,44)^$C(8,32,8)
C-VT100=80^#, $C(27,91,72,27,91,74)^24^$C(8)^W $C(27,91)_-(DY+1)_;"_-(DX+1)_-
"H" S $X=DX, $Y=DY^$C(27,91,74)^$C(27,91,75)^^^
C-VT101W=132^#, $C(27,91,72,27,91,74)^14^$C(8)^W $C(27,91)_-(DY+1)_;"_-(DX+1)_-
"H" S $X=DX, $Y=DY^$C(27,91,74)^$C(27,91,75)^^^
C-VT132=132^#, $C(27,91,72,27,91,74)^24^$C(8)^W $C(27,91)_-(DY+1)_;"_-(DX+1)_-
"H" S $X=DX, $Y=DY^$C(27,91,74)^$C(27,91,75)^^^
C-VT220=80^#, $C(27,91,72,27,91,74)^24^$C(8)^W $C(27,91)_-(DY+1)_;"_-(DX+1)_-"H"
S $X=DX, $Y=DY^$C(27,91,74)^$C(27,91,75)^$C(27,91,72,27,91,74)^$C(8,32,8)
C-VT240=80^#, $C(27,91,72,27,91,74)^24^$C(8)^W $C(27,91)_-(DY+1)_;"_-(DX+1)_-"H"
S $X=DX, $Y=DY^$C(27,91,74)^$C(27,91,75)^$C(27,91,72,27,91,74)^$C(8,32,8)
C-VT52=80^#, $C(27,72)^24^$C(8)^W $C(27,89,DY+32,DX+32) S $X=DX, $Y=DY^^^
M/UX=255^#^66^$C(8)^^^^^^
MAIL=132^#^11^$C(8)^^^^^^
```

```
P-DEC=132^#^66^$C(8)^^^^^  
PK-DEC=150^#^66^$C(8)^^^^^  
PK-QUME=150^#^66^$C(8)^^^^^
```

## Changing This Parameter

On the **Device SubTypes** page of the Management Portal (**System Administration > Configuration > Device Settings > Device SubTypes**) is a list of existing subtypes. Select **Create New Sub Type**, **Edit**, or **Delete** to modify the list.



# [ECP]

This topic describes the parameters found in the [ECP] section of the CPF.

# ClientReconnectDuration

Set duration for ECP reconnection attempt.

```
[ECP] ClientReconnectDuration=n
```

*n* is an integer in the range 10—65,636 (seconds). The default value is 1,200.

## Description

`ClientReconnectDuration` is the number of seconds an Application Server (ECP client) should keep trying to reestablish a connection before giving up or declaring the connection failed. The Application Server (ECP client) continues reconnection attempts at intervals scheduled by the [ClientReconnectInterval](#) until the full `ClientReconnectDuration` expires. The default value 1200 is equivalent to 20 minutes.

## Changing This Parameter

On the **ECP Settings** page of the Management Portal ([System Administration > Configuration > Connectivity > ECP Settings](#)), in the **This System as an ECP Application Server** column, edit **Time to wait for recovery**.

Instead of using the Management Portal, you can change `ClientReconnectDuration` in the `Config.ECP` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# ClientReconnectInterval

Set interval between ECP reconnection attempts.

```
[ECP] ClientReconnectInterval=n
```

*n* is an integer in the range 1—60 (seconds). The default value is 5.

## Description

`ClientReconnectInterval` is the number of seconds to wait between each reconnection attempt when a Data Server (ECP server) is not available. The Application Server (ECP client) continues reconnection attempts at intervals scheduled by `ClientReconnectInterval` until the full [ClientReconnectDuration](#) expires.

## Changing This Parameter

On the **ECP Settings** page of the Management Portal ([System Administration > Configuration > Connectivity > ECP Settings](#)) in the **This System as an ECP Application Server** column, edit **Time between reconnections**.

Instead of using the Management Portal, you can change `ClientReconnectInterval` in the `Config.ECP` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# ServerTroubleDuration

Set the number of seconds an ECP connection stays in troubled state.

```
[ECP] ServerTroubleDuration=n
```

*n* is an integer in the range 20—65,636 (seconds). The default value is 60.

## Description

ServerTroubleDuration is the number of seconds an ECP connection stays in a troubled state. Once this period of time has elapsed, the Data Server (ECP server) declares the connection dead and presumes that recovery is not possible.

## Changing This Parameter

On the **ECP Settings** page of the Management Portal (**System Administration > Configuration > Connectivity > ECP Settings**), in the **This System as an ECP Data Server** column, edit **Time interval for Troubled state**.

Instead of using the Management Portal, you can change ServerTroubleDuration in the Config.ECP class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# [ECP Servers]

This topic describes the server entries found in the [ECP Servers] section of the CPF.

# ECPServers

Define ECP servers.

```
[ECPServers]      Name=Address,Port, MirrorConnection
```

## Description

Entries in [ ECPServers ] define the address and port of the ECP server to connect to and the way in which connections should be redirected if the ECP server is a mirror primary. (See [Configuring Application Server Connections to a Mirror](#) in the “Mirroring” chapter of the *High Availability Guide* for important information about configuring a mirror primary as a data server.)

- *Address* – Address of the ECP server to connect to.
- *Port* – Port # of the ECP server to connect to.
- *MirrorConnection* – Behavior when connecting to a mirror primary. Default is 0 (or blank), indicating that the data server is not a mirror member. A value of 1 indicates the ECP server is a mirror failover member, and the mirror connection redirects to whichever member is primary in the event of a failover. A value of -1 indicates the ECP server is either a failover member or DR async, and the mirror connection is restricted to that specific ECP server; if the ECP server becomes the backup member, it does not accept the connection until it becomes primary.

## Changing This Parameter

On the **ECP Settings** page of the Management Portal (**System Administration > Configuration > Connectivity > ECP Settings**) is a list of ECP data servers. Select **Add Remote Data Server** to add a new ECP data server.

**Note:** You cannot set the *MirrorConnection* property to -1 from the Management Portal.

Instead of using the Management Portal, you can change `ECPServers` in the `Config.ECPServers` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# [IO]

This topic describes the parameters found in the [IO] section of the configuration parameter file.

## Other

Set the default mnemonic for WRITE commands to device types other than terminal or sequential file.

```
[IO]    Other=n
```

*n* is an InterSystems IRIS® data platform routine name. The default is ^%X364.

### Description

The **Other** setting specifies the default mnemonic for device types other than terminal or sequential file. When an [OPEN](#) or [USE](#) command includes no mnemonic space argument, InterSystems IRIS uses the default mnemonic for that device type. For more information, see the “[Controlling Devices with Mnemonic Spaces](#)” chapter in *I/O Device Guide*.

### Changing This Parameter

On the **IO Settings** page of the Management Portal ([System Administration > Configuration > Device Settings > IO Settings](#)), in the **Other** row, enter an InterSystems IRIS routine name.

Instead of using the Management Portal, you can change **Other** in the `Config.IO` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# File

Set the default mnemonic for WRITE commands to a sequential file.

```
[IO]     File=n
```

*n* is an InterSystems IRIS® data platform routine name. The default is ^%X364.

## Description

The **File** setting specifies the default mnemonic for sequential files. When an [OPEN](#) or [USE](#) command includes no mnemonic space argument, InterSystems IRIS uses the default mnemonic for that device type. For more information, see the “[Controlling Devices with Mnemonic Spaces](#)” chapter in *I/O Device Guide*.

## Changing This Parameter

On the **IO Settings** page of the Management Portal ([System Administration > Configuration > Device Settings > IO Settings](#)), in the **File** row, enter an InterSystems IRIS routine name.

Instead of using the Management Portal, you can change **File** in the [Config.IO](#) class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# Terminal

Set the default mnemonic for WRITE commands to a terminal device.

```
[IO] Terminal=n
```

*n* is an InterSystems IRIS® data platform routine name. The default is ^%X364.

## Description

The Terminal setting specifies the default mnemonic for terminal devices. When an [OPEN](#) or [USE](#) command includes no mnemonic space argument, InterSystems IRIS uses the default mnemonic for that device type. For more information, see the “[Controlling Devices with Mnemonic Spaces](#)” chapter in *I/O Device Guide*.

## Changing This Parameter

On the **IO Settings** page of the Management Portal ([System Administration > Configuration > Device Settings > IO Settings](#)), in the **Terminal** row, enter an InterSystems IRIS routine name.

Instead of using the Management Portal, you can change Terminal in the Config.IO class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# [Journal]

This topic describes the parameters found in the [Journal] section of the configuration parameter file.

# AlternateDirectory

Set the alternate location of the journal file.

```
[Journal]    AlternateDirectory=n
```

*n* is the full path to an existing directory. By default, no directory is listed.

## Description

AlternateDirectory is the name of an alternate (secondary) directory in which to store journal files.

This alternate directory is used if the primary journal directory specified by [CurrentDirectory](#) is unavailable; for example, if the disk partition for the primary journal directory is full, offline, or has some other problem. For these reasons, InterSystems recommends that the alternate journal directory be located on a different disk than the current journal directory.

When installed, the initial value of this field is “”. After InterSystems IRIS® data platform starts for the first time, the actual journal directory is filled in here, such as <*install-dir*>\mgr\journal.

## Changing This Parameter

On the **Journal Settings** page of the Management Portal ([System Administration > Configuration > System Configuration > Journal Settings](#)) , in the **Secondary journal directory** row, select **Browse**. Select the name of an existing directory.

Instead of using the Management Portal, you can change AlternateDirectory in the Config.Journal class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# BackupsBeforePurge

Set the number of backups before InterSystems IRIS® data platform purges finished journal files.

```
[Compatibility]    BackupsBeforePurge=n
```

*n* is an integer in the range 0—10. The default value is 2.

## Description

BackupsBeforePurge defines when InterSystems IRIS purges a finished journal file (that is, a journal file that is no longer in progress). The value *n* is a number of successful InterSystems IRIS system backups that must take place before the corresponding journal files can be purged.

BackupsBeforePurge relates to [DaysBeforePurge](#). If both are greater than 0, files are purged after *n* days or *n* successful backups, whichever indicates the shorter time period. If BackupsBeforePurge is 0, purging is done solely based on DaysBeforePurge; if DaysBeforePurge is 0, then purging is done solely based on BackupsBeforePurge. If both are 0, the automatic purging of journal files (and journal history) is disabled and journal files are not purged.

No journal file containing currently open transactions is purged, even if it meets the above criteria.

For details about journal files, see the “[Journaling](#)” chapter in *Data Integrity Guide*.

## Changing This Parameter

On the **Journal Settings** page of the Management Portal ([System Administration > Configuration > System Configuration > Journal Settings](#)), in the **When to purge journal files** category, choose one of the following:

- To purge journal files based on a number of days, choose **After this many days** and enter a number of days.
- To purge journal files based on a number of backups, choose **After this many successive successful backups** and enter a number of backups.

Instead of using the Management Portal, you can change BackupsBeforePurge in the Config.Journal class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# CurrentDirectory

Set the primary location of the journal file.

```
[Journal]    CurrentDirectory=n
```

*n* is the full path to an existing directory. The default is <install-dir>\mgr\journal\.

## Description

CurrentDirectory is the name of a directory in which to store the journal files (the primary directory). When installed, the initial value of this field is “”. After InterSystems IRIS® data platform starts for the first time, the actual journal directory is filled in here, such as <install-dir>\mgr\journal.

## Changing This Parameter

On the **Journal Settings** page of the Management Portal (**System Administration > Configuration > System Configuration > Journal Settings**), in the **Primary journal directory** row, select **Browse**. Select the name of an existing directory.

Instead of using the Management Portal, you can change CurrentDirectory in the Config.Journal class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DaysBeforePurge

Set the number of days before InterSystems IRIS® data platform purges finished journal files.

[Compatibility]	DaysBeforePurge=n
-----------------	-------------------

*n* is an integer in the range 0—100. The default value is 2.

## Description

**DaysBeforePurge** defines when InterSystems IRIS purges a finished journal file (that is, a journal file that is no longer in progress). The value *n* is a number days that elapse before the corresponding journal files can be purged.

**BackupsBeforePurge** relates to [DaysBeforePurge](#). If both are greater than 0, files are purged after *n* days or *n* successful backups, whichever indicates the shorter time period. If **BackupsBeforePurge** is 0, purging is done solely based on **DaysBeforePurge**; if **DaysBeforePurge** is 0, then purging is done solely based on **BackupsBeforePurge**. If both are 0, the automatic purging of journal files (and journal history) is disabled and journal files are not purged.

No journal file containing currently open transactions is purged, even if it meets the above criteria.

For details about journal files, see the “[Journaling](#)” chapter in *Data Integrity Guide*.

## Changing This Parameter

On the **Journal Settings** page of the Management Portal ([System Administration > Configuration > System Configuration > Journal Settings](#)), in the **When to purge journal files** category, choose one of the following:

- To purge journal files based on a number of days, choose **After this many days** and enter a number of days.
- To purge journal files based on a number of backups, choose **After this many successive successful backups** and enter a number of backups.

Instead of using the Management Portal, you can change **DaysBeforePurge** in the `Config.Journal` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# FileSizeLimit

Set the maximum size of a journal file.

```
[Journal]     FileSizeLimit=n
```

*n* is an integer in the range 0—4079 (MB). The default value is 1024.

## Description

`FileSizeLimit` is the maximum size of the journal file, in megabytes. When a journal file grows to this size it is closed and a new journal file is created. For more information about journal file rollover, see the [Data Integrity Guide](#).

## Changing This Parameter

On the **Journal Settings** page of the Management Portal (**System Administration > Configuration > System Configuration > Journal Settings**) , in the **Start new journal file every (MB)** row, enter a number of megabytes.

Instead of using the Management Portal, you can change `FileSizeLimit` in the `Config.Journal` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# FreezeOnError

Allow suspending InterSystems IRIS® data platform when an error occurs in writing to the journal.

```
[Compatability]    FreezeOnError=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When `FreezeOnError` is enabled (*n* = 1), then on a journal file I/O error, the Journal daemon freezes journaling immediately. The Journal daemon unfreezes journaling after it succeeds with the failed I/O operation. As soon as the error occurs all global activities that are normally journaled are blocked, which causes other jobs to hang; typically until you resolve the journaling problem. While InterSystems IRIS is hanging, you can take corrective measures, such as freeing up space on a full disk or switching the journal to a new working disk. This prevents the loss of journal data at the expense of system availability.

When `FreezeOnError` is not enabled, If `FreezeOnError=0` (false), then InterSystems IRIS does not freeze on a journal file I/O error. InterSystems IRIS disables journaling if it is not able to recover in a timely manner from the error. This prevents the system from hanging. Journaling is first retried, and then disabled, while InterSystems IRIS continues running. If journaling is disabled, you want to backup your databases as soon as possible. Running without journaling is a calculated risk, as it means the activity that occurs during this period cannot be restored. If Journaling is disabled, you MUST restart it. Ways to restart it include running the [^JRNSTART](#) routine or selecting option 1, Begin Journaling, from the [^JOURNAL](#) routine menu. See the “[Start Journaling Using ^JRNSTART](#)” section for details.

For details, see the section “[Journal I/O Errors](#)” in the “[Journaling](#)” chapter in *Data Integrity Guide*.

## Changing This Parameter

On the **Journal Settings** page of the Management Portal ([System Administration > Configuration > System Configuration > Journal Settings](#)), select `FreezeOnError` to enable this setting.

Instead of using the Management Portal, you can change `FreezeOnError` in the `Config.Journal` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “[Introduction to the Configuration Parameter File](#)” chapter in this book).

# JournalFilePrefix

Set a journal file name prefix.

```
[Journal]    JournalFilePrefix=n
```

*n* is an alphanumeric string. The default is an empty string.

## Description

`JournalFilePrefix` is a string added to the start of the journal file name. For example, if `JournalFilePrefix` is set to `Oct`, the resulting journal file would look like `Oct20191001.001`.

## Changing This Parameter

On the **Journal Settings** page of the Management Portal (**System Administration > Configuration > System Configuration > Journal Settings**), at the **Journal file prefix** row, enter a prefix.

Instead of using the Management Portal, you can change `JournalFilePrefix` in the `Config.Journal` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# JournalcspSession

Allow journaling of web sessions.

```
[Journal]    JournalcspSession=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When **JournalcspSession** is enabled (*n* = 1), InterSystems IRIS® data platform journals the %cspSession global. Enable this setting if you want the web session global to be replicated onto another machine for failover or if you want a web session to survive an InterSystems IRIS restart. Otherwise, the %cspSession global is mapped to IRISTEMP and not journaled. InterSystems IRIS kills the %cspSession global on system restart or upgrade to a new InterSystems IRIS software version, so that any record of ongoing web sessions is removed.

## Changing This Parameter

On the **Journal Settings** page of the Management Portal (**System Administration > Configuration > System Configuration > Journal Settings**), select **Journal Web session** to enable this setting.

Instead of using the Management Portal, you can change `JournalcspSession` in the `Config.Journal` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).



# [LicenseServers]

This topic describes LicenseServer, the parameter found in the [LicenseServer] section of the CPF. There can be multiple license servers defined.

# LicenseServers

Define license servers.

```
[LicenseServers]      Name=IpAddress, Port[, KeyDirectory]
```

## Description

The [LicenseServers] section contains an entry for every license server configured for InterSystems IRIS® data platform. The entries are as follows:

- *IpAddress*— IP address of the license server.
- *Port*— UDP port number used by the license server. The port numbers used at different IP addresses do not need to be different. However, the license server port number used at each IP address must be different from any UDP port number used at that IP address.
- *KeyDirectory*— License key directory. This setting is optional. If a directory is specified, on startup the License Server will read all valid .key files from that directory.

For more information about license servers, see the [Configuring InterSystems IRIS Licensing](#) section of the “Managing InterSystems IRIS Licensing” chapter in *System Administration Guide*.

## Changing This Parameter

On the **License Servers** page of the Management Portal ([System Administration > Licensing > License Servers](#)), select the name of a license server to edit it, or click **Create License Server** to add a new license server.

# [Map]

This topic describes the parameters found in the [Map] section of the CPF. It also describes parameters that may be present in sections called [Map.*w*], where *w* is the name of an InterSystems IRIS® data platform namespace.

# Global

Map globals to namespaces.

```
[Map.w]      Global_GlobalName(ss)=Database,Collation,LockLocation
```

## Description

Whenever a namespace (w) contains mappings, InterSystems IRIS® data platform creates a section in the configuration parameter (CPF) file called [Map.w] for that namespace. For example, mappings for the **USER** namespace appear under the section [Map.USER]. The [Map.w] section contains every global mapping, routine mapping, and package mapping for that namespace.

Global mappings take the form *Global\_GlobalName*, where *GlobalName* is the specific global that is being mapped. Each global entry contains three comma-separated values that map a global to a namespace. Only the first value is required. If the other values are not specified, they are set to the system default. These values are:

- *Database* — Database location of the global.
- *Collation* — Collation of the global (Default=5, InterSystems IRIS standard collation).
- *LockLocation* — Lock database location for the global (Default=Database location). InterSystems recommends that the lock database be the same as the Database location.

The collation setting is ignored if the global is not mapped at the subscript level. If the actual collation of the global does not match the defined collation in the namespace definition (for subscript level mapped globals), a <COLLATEMISMATCH> error is generated when it is referenced. This is because InterSystems IRIS requires the global to have the same collation across all the databases it lives in, regardless of the default collation for the particular database.

For more information about mappings, see the [Add Global, Routine, and Package Mapping to a Namespace](#) section of the “Configuring InterSystems IRIS” section of *System Administration Guide*.

## Subscript Mapping

You may specify subscript mapping as part of the *GlobalName* (ss). Note that if a subscript mapping is specified, a higher level mapping of the global itself must also exist. So if you want to create a mapping ^X(9), then a mapping for ^X must also exist. Subscript mappings may take the following forms:

(1)

( "A" )

(1):(5) — from 1 up to, but not including, 5

( "A" ):( "Z" ) — from A up to, but not including, Z

(BEGIN):( "X" ) — from the beginning up to, but not including, X

( "Y" ):(END) — from Y up to the end

## Examples

Map the global ^SALES to the SALES Database:

```
Global_SALES=SALES
```

Subscript map the global ^SALES("MA") to the SALESMA database:

```
Global_SALES( "MA" )=SALESMA
```

Map the global ^ACCOUNT to the database ACCOUNTS:

```
Global_ACCOUNT=ACCOUNTS
```

Subscript map the global ^ACCOUNT(1) up to but not including ACCOUNT(5) to database ACCOUNTS1TO4:

```
Global_ACCOUNT(1):(5)=ACCOUNTS1TO4
```

Map all globals starting with ABC to database ABC:

```
Global_ABC*=ABC
```

## Changing This Parameter

On the **Namespaces** page of the Management Portal (**System Administration > Configuration > System Configuration > Namespaces**), select **Global Mappings** for the namespace you want to explore. To add a new entry, click **New**. To edit an existing entry, select **Edit** in that entry's row.

# Package

Map packages to namespaces.

```
[Map.w]    Package_PackageName=Database
```

## Description

Whenever a namespace (*w*) contains mappings, InterSystems IRIS® data platform creates a section in the configuration parameter (CPF) file called [Map.w] for that namespace. For example, mappings for the **USER** namespace appear under the section [Map.USER]. The [Map.w] section contains every global mapping, routine mapping, and package mapping for that namespace.

Package mappings take the form `Package_PackageName`, where *PackageName* is the specific package that is being mapped. Each package entry contains a database location (*Database*) that contains the named package. Classes in the specified package become available in the *w* namespace.

For more information about mappings, see the [Add Global, Routine, and Package Mapping to a Namespace](#) section of the “Configuring InterSystems IRIS” section of *System Administration Guide*.

## Examples

Map package TEST to the USER database.

```
Package_TEST=USER
```

Map package TOOLS to the DEVELOPER database.

```
Package_TOOLS=DEVELOPER
```

## Changing This Parameter

On the **Namespaces** page of the Management Portal ([System Administration > Configuration > System Configuration > Namespaces](#)), select **Package Mappings** for the namespace you want to explore. To add a new entry, click **New**. To edit an existing entry, select **Edit** in that entry's row.

# Routine

---

Map routines to namespaces.

```
[Map.w] Routine_RoutineName_Type=Database
```

## Description

Whenever a namespace (*w*) contains mappings, InterSystems IRIS® data platform creates a section in the configuration parameter (CPF) file called [Map.w] for that namespace. For example, mappings for the **USER** namespace appear under the section [Map.USER]. The [Map.w] section contains every global mapping, routine mapping, and package mapping for that namespace.

Routine mappings take the form *Routine\_RoutineName*, where *RoutineName* is the specific routine that is being mapped. Each routine entry contains a database location (*Database*) that contains the named routine. The specified routine become available in the *w* namespace.

*Type* is normally not specified. *Type* only needs to be specified if you want to map part of a routine to another database. Valid values for Type are: MAC, INT, INC, or OBJ.

For more information about mappings, see the [Add Global, Routine, and Package Mapping to a Namespace](#) section of the “Configuring InterSystems IRIS” section of *System Administration Guide*.

## Examples

Map routine SALE to the SALES database:

```
Routine_SALE=SALES
```

Map all routines starting with ACC to the ACCOUNTS database:

```
Routine_ACC*=ACCOUNTS
```

Map the object code for routine TEST to the TEST database:

```
Routine_TEST_OBJ=TEST
```

## Changing This Parameter

On the **Namespaces** page of the Management Portal (**System Administration > Configuration > System Configuration > Namespaces**), select **Routine Mappings** for the namespace you want to explore. To add a new entry, click **New**. To edit an existing entry, select **Edit** in that entry's row.



# [MapMirrors]

This topic describes the [MapMirrors] section of the CPF.

# MapMirrors

Define mirror members for mirrors that include this instance.

```
[MapMirrors.m]      Name=a,b,c,d,e,f,g,h,i,j,k,l,m,n
```

## Description

For each mirror (*m*) connected to the current instance, InterSystems IRIS® data platform creates a section in the CPF called [MapMirrors.m], which contains an entry for all instances connected to that mirror. The entries are made up of a *Name* and 14 comma-separated values (*a–n*), as follows:

- *Name* – Required. Unique name identifying this member within the mirror. An uppercase alphanumeric string with a maximum length of 15 characters, cannot contain spaces or tabs, commas (,), semicolons (;), or equal signs (=), and is converted to uppercase before saving.

This is the name of this instance in this mirror. Mirror system names must be unique across all of the mirrors as an instance has a single mirror system name which may appear in multiple mirror sets (that is, an async member may connect to multiple mirrors). For the most part the *GUID* is used to identify a mirror member, the *Name* is used for display purposes. The name cannot contain a colon (:).

- *a* (AgentAddress) – The network address (IP address preferred to avoid DNS issues) that mirror members which connect to the primary should use to contact the Agent on this failover member. This is omitted on async members as the agent is not used for mirroring on those systems. This is required on failover members. The agent can transfer journal data so a private address may be desirable here to avoid network congestion.
- *b* (AgentPort) – Port # which the agent on this system is configured to listen on.  
##class(SYS.Agent).GetApplicationPort() returns the current value if the local agent is active.
- *c* – For internal use.
- *d* (SuperServerAddress) – The network address used to connect to the primary by external mirror-aware systems (currently only ECP application servers, although in the future this may extend to other connections). Other mirror members may connect to a member's superserver address for control and monitoring purposes. When a member is primary, an async member attempts to establish its data channel (over which it receives journal data) using this address if the primary's mirror private address (*MirrorPrivate*) is not accessible.
- *e* (GUID) – Required. An internal GUID, unique to this mirror. Uniquely identifies this node in the mirror. Apart from identifying the nodes, primarily used to identify the instance that owns a particular copy of a mirrored database.
- *f* (InstanceDirectory) – The installation directory of the instance (the parent of the mgr directory). Used primarily on failover members to identify the instance to the agent.
- *g* (MemberType) – Numeric value indicating the type of mirror member. One of:
  - 0 - Failover member
  - 2 - Async member
- *h* (MirrorPrivate) – When this instance is primary, other mirror members use this address to establish the mirror data channel, over which they receive journal data from the primary. Async members fall back to the primary's superserver address (*SuperServerAddress*) if they cannot reach it at the mirror private address.
- *i* (MirrorSSPort) – Superserver port for this instance. Used in conjunction with both the *MirrorAddress* and the *ECPAddress* by clients establishing connections to this instance.
- *j, k, l, m, and n* – For internal use.

## Example

Each entry is on one line:

```
[MapMirrors.MIMI]
MIMI_A=mirrorhostA,2188,,mirrorhostA,C7BA9224-3851-47D4-83BD,c:\intersystems\20142302july10a\,0,mirrorhostA,56776,,0,
MIMI_B=mirrorhostB,2188,,mirrorhostB,D14611B3-E0F5-4708-A111,c:\intersystems\20142302july10b\,0,mirrorhostB,56777,,0,
MIMI_D=mirrorhostD,2188,,mirrorhostD,06E1D307-59D9-4500-AA3B,c:\intersystems\20142302jul10d\,2,mirrorhostD,56779,,0,
```

## Changing This Parameter

On the **Create a Mirror** page of the Management Portal (**System Administration > Configuration > Mirror Settings > Create a Mirror**), enter the requested information. You can edit an existing mirror on the **Edit Mirror** page, but only from the primary failover member.



# [MirrorMember]

This topic describes the parameters found in the [MirrorMember] section of the CPF.

## AgentAddress

---

Not in use.

### Description

Not in use.

# AsyncMemberGUID

Review async member GUID

```
[MirrorMember] AsyncMemberGUID=Name
```

*Name* is any alphanumeric string.

## Description

You can create a mirror member called an *async member*, which can be configured to receive updates from one or more mirrors across the enterprise. This allows a single node to act as a comprehensive enterprise-wide data warehouse. Async members do not belong to a mirror and, therefore, are not candidates for failover.

For more information, see the [Async Mirror Members](#) section of the “Mirroring” chapter in *High Availability Guide*.

## Example

```
AsyncMemberGUID=06E1D307-59D9-4500-AA3B-4FF405E2A44D
```

## Changing This Parameter

You can change `AsyncMemberGUID` in the `Config.MirrorMember` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# AsyncMemberType

Specify async type (Disaster recovery, read-only, or read-write).

```
[MirrorMember] AsyncMemberType=n
```

*n* is either 0, 1, or 2.

## Description

AsyncMemberType indicates whether the async member is a disaster recovery (DR), read-only reporting, or read-write reporting async member.

- 0 - Disaster Recovery (DR). This is a disaster recovery async member. All its mirrored databases are read-only mirrored databases.
- 1 - Read-Only Reporting. This is a reporting async member. All its mirrored databases could be read-only or read-write databases. The default is read-only when the database is created.
- 2 - Read-Write Reporting. This is a reporting async member. All its mirrored databases could be read-only or read-write databases. The default is read-write when the database is created.

For more information, see the [Async Mirror Members](#) section of the chapter “Mirroring” in the *High Availability Guide*.

## Changing This Parameter

On the **Join as Async** page of the Management Portal (**System Administration > Configuration > Mirror Settings > Join as Async**), fill in **Mirror Information** and select **Next**. On the page **Async Member Information**, in the **Async Member System Type** row, select a type from the drop-down list.

Instead of using the Management Portal, you can change AsyncMemberType in the Config.MirrorMember class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# AsyncUseSystemPurgeInterval

Specify how mirror journal files are purged on the reporting async member.

```
[MirrorMember] AsyncUseSystemPurgeInterval=n
```

*n* is either 1 or 0.

## Description

AsyncMemberType indicates how the reporting async member purges mirror journal files received from the primary failover member.

- 0 - Mirror journal files are purged immediately after being dejournalized.
- 1 - Mirror journal files are purged according to the instances journal file purge criteria.

For more information, see the [Editing or Removing an Async Members](#) section of the chapter “Mirroring” in the *High Availability Guide*.

## Changing This Parameter

On the **Edit Async** page of the Management Portal (**System Administration > Configuration > Mirror Settings > Edit Async**), for a reporting async member, use the **Mirror Journal File Retention** drop-down to determine how mirror journal files are purged.

Instead of using the Management Portal, you can change `AsyncUseSystemPurgeInterval` in the `Config.MirrorMember` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## JoinMirror

Specify whether the instance processes its mirror configuration at startup.

```
[MirrorMember]      JoinMirror=n
```

*n* is either 1 or 0.

### Description

When `JoinMirror` is enabled ( $n = 1$ ), the mirror configuration is processed and the system is considered a mirror member according to its configuration.

When this parameter is not enabled, the mirror configuration is ignored and the system is not initialized as a mirror member. This is recommended when there is a problem in the configuration which prevents the system from starting, or if the member must be reconfigured before joining the mirror. For example, if a system was the primary but no longer is, prevents the system from joining the mirror when it restarts and attempting to become the primary again, which could result in dual primaries.

### Changing This Parameter

You can change `JoinMirror` in the `Config.MirrorMember` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# SystemName

Set the name of the mirror member.

```
[MirrorMember] SystemName=Name
```

*n* is a string with a maximum length of 32 characters.

## Description

SystemName is the name for the failover member you are configuring on this system. This defaults to a combination of the system host name and the InterSystems IRIS® data platform instance name.

Mirror member names are converted to uppercase before storing. They cannot contain spaces, tabs, or the any of following characters

: [ ] # ; / \* = ^ ~ ,

## Changing This Parameter

On any of the [Create a Mirror](#), [Join as Failover](#), or [Join as Async](#) pages of the Management Portal ([System Administration > Configuration > Mirror Settings](#)), enter the **Mirror Member Name**.

Instead of using the Management Portal, you can change SystemName in the Config.MirrorMember class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## ValidatedMember

Specify whether the instance should join the mirror in its previous role or obtain its new role from the current primary before joining the mirror.

```
[MirrorMember]    ValidatedMember=n
```

*n* is either 1 or 0.

### Description

When `ValidatedMember` is enabled ( $n = 1$ ), the instance joins the mirror in its current role.

When this parameter is not enabled, the instance contacts the primary to obtain its current role before joining the mirror. Use this when there have been role changes within the mirror while the instance and its ISCAgent were down or unreachable. For example, if a DR async has been promoted to backup while the former backup was down, set `ValidatedMember` to 0 before restarting the instance to ensure that the former backup receives its new role of DR async from the primary before restarting the mirror.

### Changing This Parameter

You can change `ValidatedMember` in the `Config.MirrorMember` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# VirtualAddressInterface

Set the local interface which hosts the primary virtual IP address.

```
[MirrorMember]    VirtualAddressInterface=IPaddress/mask
```

*IPaddress* is a valid IP address. *mask* is a Classless Inter-Domain Routing (CIDR) mask.

## Description

You can use `VirtualAddressInterface` to configure a mirror virtual IP address (VIP) so that all external clients (language bindings, ODBC/JDBC/SQL clients, and so on) connect to the mirror through a single address. For more information, see the [Configuring a Mirror Virtual IP \(VIP\)](#) section of the chapter “Mirroring” in the *High Availability Guide*.

## Changing This Parameter

On the **Create a Mirror** page of the Management Portal (**System Administration > Configuration > Mirror Settings > Create a Mirror**) select **Use Virtual IP**. Then, enter an **IP address** and a **Mask (CIDR format)**, and select a **Network Interface**.

Instead of using the Management Portal, you can change `VirtualAddressInterface` in the `Config.MirrorMember` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).



# [Mirrors]

This topic describes the [Mirrors] section of the CPF.

# Mirrors

Define mirrors that include this instance.

```
[Mirrors]      Name=a,b,c,d,e,f,g,h,i,j|k,l,m,n, o, p
```

## Description

The [Mirrors] section contains an entry for each mirror connected to the current InterSystems IRIS® data platform instance. The entries are made up of a *Name* and 14 comma-separated values (*a*—*n*), as follows:

- *Name* – Required. Unique name by which you can identify this mirror. An uppercase alphanumeric string with a maximum length of 15 characters.
- *a* (GUID) – Required. An internal GUID, unique to this mirror.
- *b* (2) – For internal use.
- *c* (1) – For internal use.
- *d* (Defined Primary) – Used to disable mirror failover, generally for maintenance purposes. Manipulated via the nofailover option when shutting down InterSystems IRIS or the **^MIRROR** routine. This contains the mirror name of the member which must be the primary. It is cleared automatically when that node starts up and becomes the primary.
- *e* (QOSTimeout) – Quality of Service Timeout: the maximum time, in milliseconds, that a failover member waits for a response from the other failover member before taking action; also applies to the arbiter's wait for a failover member's response. The default is 8000ms; typically, deployments on physical (non-virtualized) hosts with a dedicated local network can reduce this setting if a faster response to outages is required. See [Configuring the Quality of Service \(QoS\) Timeout Setting](#) in the “Mirroring” chapter in *High Availability Guide* for more information on the QoS Timeout setting.
- *f* (0) – For internal use.
- *g* (UseSSL) – To provide security within a mirror, you can configure its nodes to use SSL/TLS. This provides for both authentication of one node to another, and for encrypted communication between nodes. To use SSL/TLS with a mirror, each member (failover or sync) uses a pair of SSL/TLS configurations, %Mirror\_Client and %Mirror\_Server. These configurations must already exist on each member when SSL/TLS is enabled for the mirror. Instructions for setting up SSL are in the section [Creating and Editing SSL/TLS Configurations for a Mirror](#) of the “Using SSL/TLS with InterSystems IRIS” chapter in *Security Administration Guide*. Values are 0 (no, default) or 1 (yes).
- *h* (VirtualAddress) – Specifies a virtual IP address. You can configure a mirror virtual IP address (VIP) so that all external clients (language bindings, ODBC/JDBC/SQL clients, and so on) connect to the mirror through a single address. This virtual IP address is automatically bound to an interface on the current primary member. To use a VIP, which requires that both failover members be on the same subnet. For more information, see the [Configuring a Mirror Virtual IP \(VIP\)](#) section of the chapter “Mirroring” in the *High Availability Guide*.
- *i* (0) – For internal use.
- *j* (ArbiterNode) – The network address of the arbiter configured for this mirror. The arbiter is an independent system hosting an ISCAgent with which the failover members of a mirror maintain continuous contact, providing them with the context needed to safely make failover decisions when they cannot communicate directly.
- *k* (ArbiterPort) – The port used by the configured arbiter's ISCAgent process (2188 by default). Appears in the same space as *j*, separated by a vertical bar.
- *l* (CompressionForFailoverMembers) – Determines whether journal data is compressed before being transmitted from the primary to the backup. Possible values are 0 (System Selected, which optimizes for response time between failover members), 1 (Uncompressed), and 2 (Compressed).

- $m$  (`CompressionForAsyncMembers`) – Determines whether journal data is compressed before being transmitted from the primary to async members. Possible values are 0 (System Selected, which optimizes for network utilization), 1 (Uncompressed), and 2 (Compressed).
- $n$  (`AllowParallelDejournaling`) – Determines the type of mirror members can run parallel dejournaling updaters. Possible values are 0 (failover and disaster recover members), 1 (failover members only), and 2 (all members).
- $o$  (`CompressionTypeForFailoverMembers`) – Determines the compression type for *FailoverCompressionMode*. Possible values are 0 (ZLIB), 1 (ZSTD), and 2 (LZ4).
- $p$  (`CompressionTypeForAsyncMembers`) – Determines the compression type for *AsyncCompressionMode*. Possible values are 0 (ZLIB), 1 (ZSTD), and 2 (LZ4).

For more information on Mirroring, see the “[Mirroring](#)” chapter in *High Availability Guide*.

## Management Portal

On the **Create a Mirror** page of the Management Portal ([System Administration > Configuration > Mirror Settings > Create a Mirror](#)), enter the requested information. You can edit an existing mirror on the **Edit Mirror** page, but only from the primary failover member.



# [Miscellaneous]

This topic describes the parameters found in the [Miscellaneous] section of the CPF.

# AsyncDisconnectErr

Allow processes to receive disconnect errors asynchronously.

```
[Miscellaneous]    AsyncDisconnectErr=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

`AsyncDisconnectErr` modifies the behavior of InterSystems IRIS® data platform when `DisconnectErr` is enabled. When `AsyncDisconnectErr` is enabled (*n* = 1), the process receives an asynchronous <DSCON> error at the time a disconnect occurs on the device. This error occurs at the next command executed, and interrupts hang commands.

When this parameter is not enabled, the process receives a <DSCON> error at the next read or write command.

`AsyncDisconnectErr` is only applicable to Telnet connections on Windows. It has no effect on any other device type or operating system. If `DisconnectErr` is set to 0 (false), then `AsyncDisconnectErr` has no effect.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the `AsyncDisconnectErr` row, click **Edit**. Select `AsyncDisconnectErr` to enable this setting.

Instead of using the Management Portal, you can change `AsyncDisconnectErr` in the `Config.Miscellaneous` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the `AsyncDisconnectErr()` method of the `%SYSTEM.Process` class. See the class reference for details.

# AsynchError

Allow processes to receive asynchronous errors.

```
[Miscellaneous] AsynchError=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When **AsynchError** is enabled (*n* = 1), InterSystems IRIS® data platform processes can receive asynchronous errors.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **AsynchError** row, click **Edit**. Select **AsynchError** to enable this setting.

Instead of using the Management Portal, you can change **AsynchError** in the **Config.Miscellaneous** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **AsynchError()** method of the **%SYSTEM.Process** class. See the class reference for details.

# BreakMode

Specify programmer mode response to the BREAK command.

```
[Miscellaneous]      BreakMode=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

InterSystems IRIS® data platform programs can execute in two modes, depending on how InterSystems IRIS is entered: [application mode](#) and [programmer mode](#).

BreakMode controls how an InterSystems IRIS process in programmer mode responds when it encounters a [BREAK](#) command that has no argument. When BreakMode is enabled (*n* = 1), InterSystems IRIS enters the debugger or returns to the direct mode prompt with a <BREAK> error. When this parameter is not enabled, the BREAK command is ignored.

Application mode jobs always ignore argumentless BREAK commands.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **BreakMode** row, click **Edit**. Select **BreakMode** to enable this setting.

Instead of using the Management Portal, you can change BreakMode in the Config.Miscellaneous class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **BreakMode()** method of the %SYSTEM.Process class. See the class reference for details.

# CollectResourceStats

Allow InterSystems IRIS® data platform to collect system resource statistics.

```
[Miscellaneous]   CollectResourceStats=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

When `CollectResourceStats` is enabled (*n* = 1), InterSystems IRIS collects system resource statistics (seize, nseize, aseize, bseize).

For more information on system resource statistics, see the “[Gathering Global Activity Statistics Using ^GLOSTAT](#)” chapter and the `Enumresource` function in the [Monitoring InterSystems IRIS Using Web Services](#) appendix of *Monitoring Guide*.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the `CollectResourceStats` row, click **Edit**. Select `CollectResourceStats` to enable this setting.

Instead of using the Management Portal, you can change `CollectResourceStats` in the `Config.Miscellaneous` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DisconnectErr

Specify how processes respond to a disconnect.

```
[Miscellaneous]    DisconnectErr=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

`DisconnectErr` determines how InterSystems IRIS® data platform responds to a disconnect of the principal I/O device. When the parameter is enabled, the process receives a `<DSCON>` error when a disconnect is detected during an ObjectScript Write or Read command. When the parameter is not enabled, the process exits without reporting an error to the application when a disconnect is detected.

If `DisconnectErr` is enabled, a process continues to execute after its principal device has been disconnected. It is the responsibility of the application to detect the `<DSCON>` error and exit gracefully. Use care when enabling `DisconnectErr`.

`DisconnectErr` 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 iristerm (TRM:) connections. `DisconnectErr` is only applicable to the principal device.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **DisconnectErr** row, click **Edit**. Select **DisconnectErr** to enable this setting.

Instead of using the Management Portal, you can change `DisconnectErr` in the `Config.Miscellaneous` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **DisconnectErr()** method of the `%SYSTEM.Process` class. See the class reference for details.

# FileMode

Allow writing to a non-existent file.

```
[Miscellaneous] FileMode=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

When FileMode is enabled (*n* = 1), if a file that does not exist is opened for reading or writing, a new file is created. When this parameter is not enabled, a new file is not created (unless specified in the OPEN command parameters).

Suppose InterSystems IRIS® data platform encounters an OPEN command such as:

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

When FileMode=1 the new file is created automatically, even though the “N” parameter is not specified with the OPEN command. The result when FileMode=1 is equivalent to adding the N parameter to each OPEN command, so that the above OPEN command is equivalent to:

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

On the other hand, when InterSystems IRIS encounters an OPEN command *and* no N parameter is provided *and* the file does not already exist, then if FileMode=0 there is no result from the OPEN command except that the process hangs until interrupted.

## Changing This Parameter

On the **Compatibility** page of the Management Portal (**System Administration > Configuration > Additional Settings > Compatibility**), in the **FileMode** row, click **Edit**. Select **FileMode** to enable this setting.

Instead of using the Management Portal, you can change FileMode in the Config.Miscellaneous class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **FileMode()** method of the %SYSTEM.Process class. See the class reference for details.

# GlobalKillEnabled

Allow KILL of an unsubscribed global.

```
[Miscellaneous] GlobalKillEnabled=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

Deprecated. When **GlobalKillEnabled** is enabled (*n* = 1), a KILL of an unsubscribed global is allowed, so you can kill all subscripts of a global with a single kill instead if killing them individually. When this parameter is not enabled, the KILL results in a <PROTECT> error

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **GlobalKillEnabled** row, click **Edit**. Select **GlobalKillEnabled** to enable this setting.

Instead of using the Management Portal, you can change **GlobalKillEnabled** in the `Config.Miscellaneous` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **GlobalKillDisabled()** method of the `%SYSTEM.Process` class. Note that this is the inverse of `GlobalKillEnabled`. See the class reference for details.

# IEEEError

Specify whether \$DOUBLE returns INF and NAN values system-wide.

```
[Miscellaneous]    IEEEError=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

`IEEEError` sets the \$DOUBLE function return-value behavior system-wide. When `IEEEError` is enabled (*n* = 1), \$DOUBLE generates InterSystems IRIS® data platform errors for unresolvable IEEE floating point conversions. When this parameter is not enabled, \$DOUBLE returns INF (infinity), -INF, and NAN (Not A Number) for unresolvable IEEE floating point conversions.

The parameter 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.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **IEEEError** row, click **Edit**. Select **IEEEError** to enable this setting.

Instead of using the Management Portal, you can change `IEEEError` in the `Config.Miscellaneous` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the `IEEEError()` method of the `%SYSTEM.Process` class. See the class reference for details.

# LineRecall

Allow command line recall for READ commands.

```
[Miscellaneous]    LineRecall=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When `LineRecall` is enabled (*n* = 1), both READ commands *and* command prompts can use the line recall feature. When not enabled, only command prompts can use line recall.

## Changing This Parameter

On the **Compatibility** page of the Management Portal (**System Administration > Configuration > Additional Settings > Compatibility**), in the **LineRecall** row, click **Edit**. Select **LineRecall** to enable this setting.

Instead of using the Management Portal, you can change `LineRecall` in the `Config.Miscellaneous` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the `LineRecall()` method of the `%SYSTEM.Process` class. See the class reference for details.

# LogRollback

Allow logging for transaction rollbacks.

```
[Miscellaneous] LogRollback=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

When LogRollback is enabled (*n* = 1), InterSystems IRIS® data platform logs transaction rollbacks to the messages.log file (located in the *install-dir\mgr* directory, or the alternate directory named by the [console](#) parameter). When LogRollback is not enabled, transaction rollbacks are not logged.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **LogRollback** row, click **Edit**. Select **LogRollback** to enable this setting.

Instead of using the Management Portal, you can change LogRollback in the Config.Miscellaneous class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **LogRollback()** method of the %SYSTEM.Process class. See the class reference for details.

## MVDefined

---

Not in use.

### Description

Not in use.

# NodeNameInPid

Specify behavior when InterSystems IRIS® data platform references the special variable \$JOB.

[Miscellaneous]      **NodeNameInPid=n**

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

When `NodeNameInPid` is enabled (*n* = 1), `$JOB` returns the process ID number of the current process concatenated to the node name. When `NodeNameInPid` is not enabled, `$JOB` returns only the process ID number, without the node name.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **NodeNameInPid** row, click **Edit**. Select **NodeNameInPid** to enable this setting.

Instead of using the Management Portal, you can change `NodeNameInPid` in the `Config.Miscellaneous` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the `NodeNameInPid()` method of the `%SYSTEM.Process` class. See the class reference for details.

# NullSubscripts

Allow null subscripts on global references.

```
[Miscellaneous]    NullSubscripts=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

When **NullSubscripts** is enabled (*n* = 1), null subscripts are allowed on global references. When this parameter is not enabled, a null subscript causes a <SUBSCRIPT> error. InterSystems recommends leaving this setting disabled.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **NullSubscripts** row, click **Edit**. Select **NullSubscripts** to enable this setting.

Instead of using the Management Portal, you can change **NullSubscripts** in the **Config.Miscellaneous** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **NullSubscripts()** method of the **%SYSTEM.Process** class. See the class reference for details.

# OldZU5

Specify whether to clear global vectors when switching namespace.

```
[Miscellaneous]    OldZU5=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

When `OldZU5` is enabled (*n* = 1), switching to the current namespace using the `ZN` command clears the global vector cache. When this parameter is not enabled, switching to the current namespace has no effect.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **OldZU5** row, click **Edit**. Select **OldZU5** to enable this setting.

Instead of using the Management Portal, you can change `OldZU5` in the `Config.Miscellaneous` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the `OldZU5()` method of the `%SYSTEM.Process` class. See the class reference for details.

# OpenMode

Specify read/write mode to use when opening sequential files.

```
[Miscellaneous]      OpenMode=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

OpenMode specifies the default read/write mode to use when opening sequential files with the [OPEN](#) command. The options are **Read-Write** (1) or **Read** (0).

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **OpenMode** row, click **Edit**. Choose a mode, **Read** (0) or **Read-Write** (1).

Instead of using the Management Portal, you can change OpenMode in the Config.Miscellaneous class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **OpenMode()** method of the %SYSTEM.Process class. See the class reference for details.

# PopError

Specify when to pop error handlers off the stack.

```
[Miscellaneous]      PopError=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

When **PopError** is enabled (*n* = 1), InterSystems IRIS® data platform pop the \$ZTRAP error handler off the stack when an error is triggered. In this case, when a \$ZTRAP error handler is invoked by the system, 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.

When this parameter is not enabled, the normal behavior prevails: A \$ZTRAP error handler stays active when the error handler is invoked. In this case, 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.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **PopError** row, click **Edit**. Select **PopError** to enable this setting.

Instead of using the Management Portal, you can change **PopError** in the `Config.Miscellaneous` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **PopError()** method of the `%SYSTEM.Process` class. See the class reference for details.

# RefInKind

Specify how \$NAME and \$QUERY handle extended global references.

[Miscellaneous] RefInKind=*n*

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

When RefInKind is enabled (*n* = 1), given an input that is an [extended global reference](#), \$NAME and \$QUERY return only the global name without the extended reference. When this parameter is not enabled, the functions return an extended global reference.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **RefInKind** row, click **Edit**. Select **RefInKind** to enable this setting.

Instead of using the Management Portal, you can change RefInKind in the Config.Miscellaneous class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

**Note:** 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.

To change this parameter for a single process only (as opposed to system-wide), use the **RefInKind()** method of the %SYSTEM.Process class. See the class reference for details.

# ScientificNotation

Allow lowercase "e" as scientific notation symbol system-wide.

[Miscellaneous]      ScientificNotation=*n*

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When **RefInKind** is enabled (*n* = 1), InterSystems IRIS® data platform uses the lowercase "e" as scientific notation symbol system-wide.

## Changing This Parameter

On the **Compatibility** page of the Management Portal (**System Administration > Configuration > Additional Settings > Compatibility**), in the **ScientificNotation** row, click **Edit**. Select **ScientificNotation** to enable this setting.

Instead of using the Management Portal, you can change **ScientificNotation** in the **Config.Miscellaneous** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **ScientificNotation()** method of the **%SYSTEM.Process** class. See the class reference for details.

# SetZEOF

Specify the behavior when reading a sequential file and encountering an unexpected end-of-file error.

```
[Miscellaneous] SetZEOF=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

When SetZEOF is enabled (*n* = 1), InterSystems IRIS® data platform sets the special variable \$ZEOF to indicate that you have reached the end of a sequential file. When this parameter is not enabled, InterSystems IRIS throws an <ENDOFFILE> error instead.

## Changing This Parameter

On the **Compatibility** page of the Management Portal (**System Administration > Configuration > Additional Settings > Compatibility**), in the **SetZEOF** row, click **Edit**. Select **SetZEOF** to enable this setting.

Instead of using the Management Portal, you can change SetZEOF in the Config.Miscellaneous class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **SetZEOF()** method of the %SYSTEM.Process class. See the class reference for details.

# ShutDownLogErrors

Allow writing of InterSystems IRIS® data platform system error log entries to the messages.log file on shutdown.

[Miscellaneous]	ShutDownLogErrors=n
-----------------	---------------------

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

When `ShutDownLogErrors` is enabled ( $n = 1$ ), during shutdown InterSystems IRIS logs error information from `^SYSLOG` to the `messages.log` file (located in the `install-dir\mgr` directory, or the alternate directory named by the `console` parameter). When `ShutDownLogErrors` is not enabled, these errors are not logged.

For more information, see [InterSystems IRIS System Error Log](#) in the “Monitoring InterSystems IRIS Using the Management Portal” chapter in *Monitoring Guide*.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **ShutDownLogErrors** row, click **Edit**. Select **ShutDownLogErrors** to enable this setting.

Instead of using the Management Portal, you can change `ShutDownLogErrors` in the `Config.Miscellaneous` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## StopID

---

Not in use.

## Description

Not in use.

# SwitchOSdir

Disallow switching current working directories when changing namespaces.

[Miscellaneous]	SwitchOSdir=n
-----------------	---------------

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

SwitchOSdir specifies what happens to the current working directory (for accessing files by relative pathname, etc.) when you switch to a different namespace. When SwitchOSdir is enabled (*n* = 1), if you change namespaces, the current working directory remains unaltered no matter what namespace you switch to.

When this parameter is not enabled, 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.

For example, suppose SwitchOSdir is set to 1, *or* SwitchOSdir is set to 0 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.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **SwitchOSDir** row, click **Edit**. Select **SwitchOSDir** to enable this setting.

Instead of using the Management Portal, you can change SwitchOSDir in the Config.Miscellaneous class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **SwitchOSDir()** method of the %SYSTEM.Process class. See the class reference for details.

# SynchCommit

Disable synchronizing TCOMMIT with the corresponding journal write operation.

```
[Miscellaneous]      SynchCommit=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

Every **TCOMMIT** command requests a flush of the journal data involved in that transaction to disk. **SynchCommit** controls what happens at that point. When enabled (*n* = 1), TCOMMIT completes after the journal data write operation completes. If **SynchCommit** is not enabled, TCOMMIT completes without waiting for the write operation.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **SynchCommit** row, click **Edit**. Select **SynchCommit** to enable this setting.

Instead of using the Management Portal, you can change **SynchCommit** in the **Config.Miscellaneous** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **SynchCommit()** method of the **%SYSTEM.Process** class. See the class reference for details.

# TelnetNUL

Suppress Telnet NUL at end-of-line for Telnet transmission. Windows systems only.

```
[Miscellaneous]    TelnetNUL=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

On output, a Telnet network virtual terminal (NVT) performs the following default end-of-line behavior: either issues a carriage return character (CR) followed by a linefeed character (LF), or issues a CR followed by NUL (if no LF is issued). TelnetNUL affects the issuance of the NUL character in the second case. When TelnetNUL is enabled (*n* = 1), the Telnet virtual terminal suppresses the NUL character.

This setting applies only to Windows systems; it is ignored on UNIX®, and Linux configurations, in which Telnet is supplied by the operating system vendor.

## Changing This Parameter

On the **Compatibility** page of the Management Portal (**System Administration > Configuration > Additional Settings > Compatibility**), in the **TelnetNUL** row, click **Edit**. Select **TelnetNUL** to enable this setting.

Instead of using the Management Portal, you can change TelnetNUL in the Config.Miscellaneous class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **TelnetNUL()** method of the %SYSTEM.Process class. See the class reference for details.

# TruncateOverflow

Suppress the <MAXNUMBER> error on numeric overflow.

[Miscellaneous]	TruncateOverflow=n
-----------------	--------------------

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

Normally, when InterSystems IRIS® data platform encounters an extremely large number (on the order of 1.0E147, or -1.0E146), it throws the <MAXNUMBER> error. When TruncateOverflow is enabled, the <MAXNUMBER> error is suppressed.

For more information, see the [Extremely Large Numbers](#) section of the “Data Types and Values” chapter in *Using ObjectScript*.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **TruncateOverflow** row, click **Edit**. Select **TruncateOverflow** to enable this setting.

Instead of using the Management Portal, you can change TruncateOverflow in the Config.Miscellaneous class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **TruncateOverflow()** method of the %SYSTEM.Process class. See the class reference for details.

## See Also

- [\\$DOUBLE](#) function in *ObjectScript Reference*.

# Undefined

Specify the response when ObjectScript attempts to fetch a variable that does not exist.

[Miscellaneous]      **Undefined=n**

*n* is either 0, 1, or 2. The default value is 0.

## Description

The parameter **Undefined** specifies the behavior when ObjectScript attempts to fetch the value of a variable that has not been defined. The value of **Undefined** may be 0, 1, or 2:

- 0 - Always 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.

## Changing This Parameter

On the **Compatibility** page Management Portal (**System Administration > Configuration > Additional Settings > Compatibility**), in the **Undefined** row, click **Edit**. Select the option you want.

Instead of using the Management Portal, you can change **Undefined** in the **Config.Miscellaneous** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **Undefined()** method of the **%SYSTEM.Process** class. See the class reference for details.

# UseNagleAlgorithm

Allow InterSystems IRIS® data platform to use the Nagle algorithm for Telnet.

[Miscellaneous]	UseNagleAlgorithm= <i>n</i>
-----------------	-----------------------------

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

When **UseNagleAlgorithm** is enabled (*n* = 1), InterSystems IRIS uses 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. 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 calls send again with more data that can be consolidated with the first. For more details see [RFC 896](#).

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **UseNagleAlgorithm** row, click **Edit**. Select **UseNagleAlgorithm** to enable this setting.

Instead of using the Management Portal, you can change **UseNagleAlgorithm** in the **Config.Miscellaneous** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# ViewPastData

Allow \$VIEW to examine data outside of InterSystems IRIS® data platform memory area.

```
[Miscellaneous]      ViewPastData=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

When ViewPastData is enabled (*n* = 1), you can use the [\\$VIEW](#) command to examine data outside of InterSystems IRIS memory area. When this parameter is not enabled, the \$VIEW command throws an error.

## Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **ViewPastData** row, click **Edit**. Select **ViewPastData** to enable this setting.

Instead of using the Management Portal, you can change ViewPastData in the Config.Miscellaneous class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## ZDateNull

Specify the \$ZDATE response to an invalid value.

```
[Miscellaneous] ZDateNull=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

### Description

ZDateNull determines how a [\\$ZDATE](#) call responds when triggered by an invalid value. When this parameter is enabled, \$ZDATE returns a null value. When this parameter is not enabled, it returns an error.

### Changing This Parameter

On the **Compatibility** page of the Management Portal ([System Administration > Configuration > Additional Settings > Compatibility](#)), in the **ZDateNull** row, click **Edit**. Select **ZDateNull** to enable this setting.

Instead of using the Management Portal, you can change ZDateNull in the Config.Miscellaneous class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

To change this parameter for a single process only (as opposed to system-wide), use the **ZDateNull()** method of the %SYSTEM.Process class. See the class reference for details.

# [Monitor]

This topic describes the parameters found in the [Monitor] section of the CPF.

## SNMPEnabled

Allow automatic Simple Network Management Protocol (SNMP) startup.

```
[Monitor]    SNMPEnabled=n
```

*n* is either 1 or 0. The default value is 0.

### Description

When `SNMPEnabled` is enabled (*n* = 1), the SNMP agent automatically starts when InterSystems IRIS® data platform starts up.

To enable SNMP monitoring, select **Start SNMP Agent at System Startup**. You must also have the `%Service_Monitor` enabled on the **Services** page (**System Administration > Security > Services**). For more information, see [Monitoring InterSystems IRIS Using SNMP](#) chapter in *Monitoring Guide*.

### Changing This Parameter

On the **Monitor** page of the Management Portal (**System Administration > Configuration > Additional Settings > Monitor**), select **Start SNMP Agent at System Startup** to enable this setting. There is also a shortcut to enable `%Service_Monitor`.

Instead of using the Management Portal, you can change `SNMPEnabled` in the `Config.Monitor` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# [Namespaces]

This topic describes the parameters found in the [Namespaces] section of the CPF.

# Namespace

Define InterSystems IRIS® data platform namespaces.

[Namespaces]	Name=global,routines,temporary
--------------	--------------------------------

## Description

The [Namespaces] section contains an entry for every namespace defined for the InterSystems IRIS instance. InterSystems IRIS adds entries to the configuration parameter file automatically as you add and configure namespaces using the Management Portal.

Each entry contains up to three comma-separated values, but only the first value is required. If the other values are not specified, they are set to the system default. The values are as follows:

- *globals* — Default database name for globals (other than temporary globals). Required.
- *routines* — Default database name for routines and classes. If the database is not specified, it defaults to the globals database.
- *temporary* — Default database name for temporary storage, specifically storage of temporary globals. If the database is not specified, it defaults to IRISTEMP.

A *temporary global* is a global whose name starts with ^IRIS.Temp (case-sensitive).

## Exceptions to Defaults

Globals starting with a % are mapped to IRISSYS unless mapped to another database by a user-defined [global](#) mapping.

In all namespaces, routine and classes that start with a % come from the IRISLIB database with the following exceptions:

- Routines and classes starting with %SYS.\* come from IRISSYS (supplied by InterSystems IRIS).
- Routines and classes starting with %Z\* and %z\* come from IRISSYS (user defined routines and classes).
- Routines and classes that are explicitly mapped from another database by the user using [routine](#) or [package](#) mapping.

Note that routines that reside in the IRISSYS database have special security privileges including the ability to modify the roles and other security attributes of the process executing them.

## Examples

In the [Namespaces] section, each entry appears all on one line:

```
[Namespaces]
%SYS=IRISSYS
USER=USER
SALES=SALESGBL,SALESRTN
; Globals and routines/classes split into separate databases.
BILLING=BILLING,,TEMPDATA
; Globals and routines/classes in the same database,
; temporary globals are mapped to the databases TEMPDATA
```

## Changing This Parameter

On the **Namespaces** page of the Management Portal ([System Administration > Configuration > System Configuration > Namespaces](#)), to add a new entry, select **Create New Namespace**. To edit an existing entry, select **Edit** in that entry's row.

## See Also

The [Global](#) entry in the [Map] section of this book.

The [Package](#) entry in the [Map] section of this book.

The [Routine](#) entry in the [Map] section of this book.



# [SQL]

This topic describes the parameters found in the [SQL] section of the CPF.

# AllowRowIDUpdate

Allow user to update RowID values.

```
[SQL]      AllowRowIDUpdate=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When `AllowRowIDUpdate` is enabled (*n* = 1), RowID values are user-modifiable. Modifying RowID values can have serious consequences and should only be done in very specific cases and with extreme caution. Set to 1 only if you are doing your own filing in a BEFORE trigger and using the `%SkipFiling` flag. Otherwise, use the default of 0.

## Changing This Parameter

You can change `AllowRowIDUpdate` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

- [RowID Field](#) section in “Defining Tables” chapter in *Using InterSystems SQL*.

# ANSIPrecedence

Specify operator precedence for SQL queries.

```
[SQL]      ANSIPrecedence=n
```

*n* is either 1 or 0. The default value is 1.

## Description

When ANSIPrecedence is enabled (*n* = 1), InterSystems SQL uses ANSI precedence of arithmetic operators. If ANSIPrecedence is disabled (*n* = 0), InterSystems SQL executes arithmetic expressions in strict left-to-right order. This is a system-wide configuration setting.

When ANSI precedence is configured, the “\*”, “\”, “/”, and “#” operators have a higher precedence than the “+”, “-”, and “||” operators. Operators with a higher precedence are executed before operators with a lower precedence. You can use parentheses to override precedence when desired.

For further details, refer to the [Operator Precedence](#) section of the *Using InterSystems SQL Guide*.

## Changing This Parameter

To set the desired value for ANSIPrecedence from the InterSystems IRIS® data platform Terminal, use the [SetANSIPrecedence\(\)](#) method of the %SYSTEM.SQL class. See the class reference for details.

You can also change ANSIPrecedence with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

- [Arithmetic Operators and Functions](#) section of the “Language Elements” chapter in *Using InterSystems SQL Guide*.

# AutoParallel

Allow parallel processing system-wide.

```
[SQL] AutoParallel=n
```

*n* is either 1 or 0. The default value is 1.

## Description

When AutoParallel is enabled (*n* = 1), InterSystems SQL queries can use [parallel processing](#) to run more efficiently. In sharded environments, this means all queries are executed using parallel processing. In non-sharded environments, InterSystems SQL determines per-query whether to use parallel processing, based on the value of [AutoParallelThreshold](#).

When this parameter is not enabled, all queries run in a single process.

## Changing This Parameter

On the **SQL** page of the Management Portal (**System Administration > Configuration > SQL and Object Settings > SQL**), select **Execute queries in a single process** to disable parallel processing.

To set the desired value for AutoParallel from the InterSystems IRIS® data platform Terminal, use the **SetAutoParallel()** method of the %SYSTEM.SQL class. See the class reference for details.

You can also change AutoParallel with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# AutoParallelThreshold

Set the threshold for parallel processing.

```
[SQL] AutoParallelThreshold=n
```

*n* is any nonnegative integer. The default value is 3200.

## Description

The higher *n* is, the lower the chance that an InterSystems SQL query executes using [parallel processing](#). The value *n* corresponds roughly to the minimal number of tuples needed in the visited map for parallel processing to occur.

When [AutoParallel](#) is disabled, AutoParallelThreshold has no effect.

## Changing This Parameter

To set the desired value for AutoParallelThreshold from the InterSystems IRIS® data platform Terminal, use the [SetAutoParallelThreshold\(\)](#) method of the %SYSTEM.SQL class. See the class reference for details.

You can also change AutoParallelThreshold with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## BiasQueriesAsOutlier

Set query optimization to biased toward outliers.

```
[SQL] BiasQueriesAsOutlier=n
```

*n* is either 1 or 0. The default value is 0.

### Description

When `BiasQueriesAsOutlier` is enabled (*n* = 1), InterSystems SQL optimizes for queries that primarily return [outlier values](#). For further details on outlier selectivity, refer to [Tune Table](#) in the “Optimizing Tables” chapter in *InterSystems SQL Optimization Guide*.

`BiasQueriesAsOutlier` cannot be set to 1 if the parameter `RTPC` is also set to 1. When `RTPC` is set, InterSystems SQL determines whether to use outlier optimization on a per-query basis.

### Changing This Parameter

On the `SQL` page of the Management Portal ([System Administration > Configuration > SQL and Object Settings > SQL](#)), for the **Optimize queries based on parameter values** setting, choose **Assume query parameter values often match field outliers**.

You can also change `BiasQueriesAsOutlier` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# Comment

Retain embedded SQL statements as comments in source code.

```
[SQL]      Comment=n
```

*n* is either 1 or 0. The default value is 1.

## Description

When Comment is enabled (*n* = 1), embedded SQL statements are retained as comments in the source code (.INT) version of the routine.

## Changing This Parameter

To set the desired value for Comment from the InterSystems IRIS® data platform Terminal, use the **SetRetainSQL()** method of the %SYSTEM.SQL class. See the class reference for details.

You can also change Comment with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DBMSSecurity

Enable SQL security.

```
[SQL]      DBMSSecurity=n
```

*n* is either 1 or 0. The default value is 1.

## Description

When `DBMSSecurity` is enabled, all InterSystems SQL security is enabled. This means privilege-based table/view/procedure security is active. A user can only view or perform actions on a table for which that user has been granted privilege. When this parameter is not enabled, a user can view or perform actions on a table even if that user lacks the necessary privilege.

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

## Changing This Parameter

To set the desired value for `DBMSSecurity` from the InterSystems IRIS® data platform Terminal, use the [`SetSQLSecurity\(\)`](#) method of the `%SYSTEM.SQL` class. See the class reference for details. Changing this option affects only new processes; existing processes are not affected.

You can also change `DBMSSecurity` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DDLDefineBitmapExtent

Specify whether a table created by a DDL statement defines a bitmap extent index.

```
[SQL]      DDLDefineBitmapExtent=n
```

*n* is either 1 or 0. The default value is 1.

## Description

When `DDLDefineBitmapExtent` is enabled (*n* = 1), a table created by a DDL CREATE TABLE statement defines a bitmap extent index. The index improves the performance of `COUNT(*)`, a function that returns the number of rows in the table.

## Changing This Parameter

To set the desired value for `DDLDefineBitmapExtent` from the InterSystems IRIS® data platform Terminal, use the `SetDDLDefineBitmapExtent()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `DDLDefineBitmapExtent` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## DDLFinal

Specify whether a class created by a DDL statement is final.

```
[SQL]      DDLFinal=n
```

*n* is either 1 or 0. The default value is 1.

### Description

When `DDLFinal` is enabled (*n* = 1), a class created by a DDL CREATE TABLE statement is [final](#), meaning it cannot have subclasses.

### Changing This Parameter

To set the desired value for `DDLFinal` from the InterSystems IRIS® data platform Terminal, use the `SetDDLFinal()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `DDLFinal` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DDLNo201

Suppress error upon CREATE of a previously existing table.

```
[SQL]      No201=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When DDLNo201 is enabled (*n* = 1), when an attempt is made to CREATE a previously existing table or view, InterSystems IRIS® data platform suppresses the SQLCODE -201 error. When this parameter is not enabled, InterSystems IRIS returns the error.

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

## Changing This Parameter

On the **SQL** page of the Management Portal (**System Administration > Configuration > SQL and Object Settings > SQL**), select **Ignore redundant DDL statements** to suppress redundant SQLCODE errors.

To set the desired value for DDLNo201 from the InterSystems IRIS Terminal, use the **SetDDLNo201()** method of the **%SYSTEM.SQL** class. See the class reference for details.

You can also change DDLNo201 with the **Config.SQL** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## DDLNo30

Suppress error upon DROP of a nonexistent table.

```
[SQL]      DDLNo30=n
```

*n* is either 1 or 0. The default value is 0.

### Description

When DDLNo30 is enabled (*n* = 1), when an attempt is made to DROP a nonexistent table, InterSystems IRIS® data platform suppresses the SQLCODE -30 error. When this parameter is not enabled, InterSystems IRIS returns the error.

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

### Changing This Parameter

On the **SQL** page of the Management Portal ([System Administration > Configuration > SQL and Object Settings > SQL](#)), select **Ignore redundant DDL statements** to suppress redundant SQLCODE errors.

To set the desired value for DDLNo30 from the InterSystems IRIS Terminal, use the **SetDDLNo30()** method of the %SYSTEM.SQL class. See the class reference for details.

You can also change DDLNo30 with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DDLNo307

Suppress error upon CREATE of a primary key constraint when one exists.

```
[SQL]      DDLNo307=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When DDLNo307 is enabled (*n* = 1), when an attempt is made to CREATE a primary key constraint to a table through DDL, and a primary key constraint already exists for that table, InterSystems IRIS® data platform suppresses the SQLCODE -307 error.

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

## Changing This Parameter

On the **SQL** page of the Management Portal (**System Administration > Configuration > SQL and Object Settings > SQL**), select **Ignore redundant DDL statements** to suppress redundant SQLCODE errors.

To set the desired value for DDLNo307 from the InterSystems IRIS Terminal, use the **SetDDLNo307()** method of the **%SYSTEM.SQL** class. See the class reference for details.

You can also change DDLNo307 with the **Config.SQL** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DDLNo311

Suppress error upon ADD a foreign key, when a key of that name already exists.

```
[SQL]      DDLNo311=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When DDLNo311 is enabled (*n* = 1), when an attempt is made to ADD a foreign key, even if a key of that name already exists, InterSystems IRIS® data platform suppresses the SQLCODE -311 error.

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

## Changing This Parameter

On the **SQL** page of the Management Portal ([System Administration > Configuration > SQL and Object Settings > SQL](#)), select **Ignore redundant DDL statements** to suppress redundant SQLCODE errors.

To set the desired value for DDLNo311 from the InterSystems IRIS Terminal, use the **SetDDLNo311()** method of the %SYSTEM.SQL class. See the class reference for details.

You can also change DDLNo311 with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DDLNo315

Suppress error upon DROP of a nonexistent constraint.

```
[SQL]      DDLNo315=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When DDLNo315 is enabled (*n* = 1), when an attempt is made to DROP a nonexistent constraint, InterSystems IRIS® data platform suppresses the SQLCODE -315 error.

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

## Changing This Parameter

On the **SQL** page of the Management Portal ([System Administration > Configuration > SQL and Object Settings > SQL](#)), select **Ignore redundant DDL statements** to suppress redundant SQLCODE errors.

To set the desired value for DDLNo315 from the InterSystems IRIS Terminal, use the **SetDDLNo315()** method of the %SYSTEM.SQL class. See the class reference for details.

You can also change DDLNo315 with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DDLNo324

Suppress error upon CREATE of a previously existing index.

```
[SQL]      DDLNo324=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When DDLNo324 is enabled (*n* = 1), when an attempt is made to CREATE a previously existing index, InterSystems IRIS® data platform suppresses the SQLCODE -324 error.

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

## Changing This Parameter

On the **SQL** page of the Management Portal ([System Administration > Configuration > SQL and Object Settings > SQL](#)), select **Ignore redundant DDL statements** to suppress redundant SQLCODE errors.

To set the desired value for DDLNo324 from the InterSystems IRIS Terminal, use the **SetDDLNo324()** method of the %SYSTEM.SQL class. See the class reference for details.

You can also change DDLNo324 with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DDLNo333

Suppress error upon DROP of a nonexistent index.

```
[SQL]      DDLNo333=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When DDLNo333 is enabled (*n* = 1), when an attempt is made to DROP a nonexistent index, InterSystems IRIS® data platform suppresses the SQLCODE -333 error.

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

## Changing This Parameter

On the **SQL** page of the Management Portal ([System Administration > Configuration > SQL and Object Settings > SQL](#)), select **Ignore redundant DDL statements** to suppress redundant SQLCODE errors.

To set the desired value for DDLNo333 from the InterSystems IRIS Terminal, use the **SetDDLNo333()** method of the %SYSTEM.SQL class. See the class reference for details.

You can also change DDLNo333 with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DDLSQLOnlyCompile

Enable an SQL-only compile.

```
[SQL] DDLSQLOnlyCompile=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When `DDLSQLOnlyCompile` is enabled (*n* = 1), any class compilation performed as a result of executing a DDL statement compiles the class with the *q* (sqlonly) flag. When this parameter is not enabled, the *q* flag is not used.

## Changing This Parameter

You can change `DDLSQLOnlyCompile` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DDLUseExtentSet

Allow hashed names for globals that store index data of tables created by a DDL statement.

```
[SQL]      DDLUseExtentSet=n
```

*n* is either 1 or 0. The default value is 1.

## Description

When `DDLUseExtentSet` is enabled (*n* = 1), a table created by a DDL CREATE TABLE statement stores its index data in globals that use hashed names. The hashed names typically allow for better performance when running queries against the table, but are less comprehensible to the user. When this parameter is not enabled, the index data is stored in globals named after the class.

## Changing This Parameter

To set the desired value for `DDLUseExtentSet` from the InterSystems IRIS® data platform Terminal, use the `SetDDLUseExtentSet()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `DDLUseExtentSet` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DDLUseSequence

Specify the function a table created by a DDL statement uses for ID assignment.

```
[SQL]      DDLUseSequence=n
```

*n* is either 1 or 0. The default value is 1.

## Description

When `DDLUseSequence` is enabled (*n* = 1), a table created by DDL CREATE TABLE uses `$SEQUENCE` for ID assignment. When this parameter is not enabled, the table uses `$INCREMENT`.

`$SEQUENCE` is the default function, and is better suited for ID assignment. For a comparison of the two functions, see the [\\$INCREMENT or \\$SEQUENCE](#) section in the chapter “`$INCREMENT`” in *ObjectScript Reference*.

## Changing This Parameter

To set the desired value for `DDLUseSequence` from the InterSystems IRIS® data platform Terminal, use the `SetDDLUseSequence()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `DDLUseSequence` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DefaultSchema

Set the default SQL schema name.

```
[SQL]      DefaultSchema=n
```

*n* is a string with a maximum length of 128 characters. The default string is SQLUser.

## Description

DefaultSchema defines the default SQL schema name. 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.

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

## Changing This Parameter

On the **SQL** page of the Management Portal (**System Administration > Configuration > SQL and Object Settings > SQL**), for the **Default Schema** setting, enter a string of characters.

To set the desired value for DefaultSchema from the InterSystems IRIS® data platform Terminal, use the **SetDefaultSchema()** method of the %SYSTEM.SQL class. See the class reference for details.

You can also change DefaultSchema with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DelimitedIds

Enable interpreting double-quoted strings as delimited identifiers.

```
[SQL]     DelimitedIds=n
```

*n* is either 1 or 0. The default value is 1.

## Description

When `DelimitedIds` is enabled (*n* = 1), a double-quoted string ("My String") is considered a delimited identifier within an SQL statement. When this parameter is not enabled, a double-quoted string ("My String") is considered a string constant or literal string.

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

## Changing This Parameter

To set the desired value for `DelimitedIds` from the InterSystems IRIS® data platform Terminal, use the `SetDelimitedIdentifiers()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `DelimitedIds` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DropDelete

Specify whether DROP TABLE deletes the table's data in addition to the table.

```
[SQL]      DropDelete=n
```

*n* is either 1 or 0. The default value is 1.

## Description

When `DropDelete` is enabled (*n* = 1), a `DROP TABLE` statement deletes the table *and* the table's data. When this parameter is not enabled, a `DROP TABLE` statement deletes the table, but does not delete the data.

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

## Changing This Parameter

To set the desired value for `DropDelete` from the InterSystems IRIS® data platform Terminal, use the `SetDDLDropTabDelData()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `DropDelete` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# ECPSync

Ensure that the server and client cache are in sync.

```
[SQL]      ECPSync=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When ECPSync is enabled (*n* = 1), each time a **SELECT** statement is executed InterSystems IRIS® data platform forces all pending Enterprise Cache Protocol (ECP) requests to the database server. On completion, this guarantees that the client cache is in sync.

ECP is a distributed data caching architecture that manages the distribution of data and locks among a heterogeneous network of server systems. For further details, refer to [Queries and ECP](#) in the “Querying the Database” chapter in *Using InterSystems SQL*.

## Changing This Parameter

To set the desired value for ECPSync from the InterSystems IRIS Terminal, use the **SetECPSync()** method of the %SYSTEM.SQL class. See the class reference for details.

You can also change ECPSync with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# ExtrinsicFunctions

Enable extrinsic functions in SQL statements.

```
[SQL]      ExtrinsicFunctions=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When `ExtrinsicFunctions` is enabled (*n* = 1), extrinsic functions can be used in SQL statements through ODBC, JDBC, and Dynamic Query.

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

## Changing This Parameter

To set the desired value for `ExtrinsicFunctions` from the InterSystems IRIS® data platform Terminal, use the `SetAllowExtrinsicFunctions()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `ExtrinsicFunctions` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# FastDistinct

Allow SQL DISTINCT optimization.

```
[SQL]      FastDistinct=n
```

*n* is either 1 or 0. The default value is 1.

## Description

When **FastDistinct** is enabled (*n* = 1), SQL queries involving DISTINCT or GROUP BY clauses run more efficiently by making better use of indices (if indices are available).

**CAUTION:** The values returned by such queries are collated in the same way they are stored within the index. This means the results of such queries may be all uppercase. This may have an effect on case-sensitive applications.

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

## Changing This Parameter

**FastDistinct** is enabled by default. On the **SQL** page of the Management Portal ([System Administration > Configuration > SQL and Object Settings > SQL](#)), select **GROUP BY and DISTINCT queries must produce original values** to disable **FastDistinct**.

To set the desired value for **FastDistinct** from the InterSystems IRIS® data platform Terminal, use the **SetFastDistinct()** method of the **%SYSTEM.SQL** class. See the class reference for details.

You can also change **FastDistinct** with the **Config.SQL** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# IdKey

Set primary key constraint behavior.

```
[SQL]      IdKey=n
```

*n* is either 1 or 0. The default value is 1.

## Description

When `IdKey` is enabled (*n* = 1), when a Primary Key constraint is specified through DDL it does *not* also become the IDKey index in the class definition.

When this parameter is not enabled, a Primary Key constraint specified through DDL also becomes the IDKey index in the class definition. This option generally gives better performance, but means that the Primary Key fields cannot be updated.

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

## Changing This Parameter

On the **SQL** page of the Management Portal ([System Administration > Configuration > SQL and Object Settings > SQL](#)), select **Define primary key as ID key for tables created via DDL** to disable `IdKey`.

To set the desired value for `IdKey` from the InterSystems IRIS® data platform Terminal, use the `SetDDLPKeyNotIDKey()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `IdKey` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## IdTrxFrom

Define the “From” list of characters for Identifier Translation.

```
[SQL]      IdTrxFrom=n
```

*n* is a string with a maximum length of 256 characters. The default string is ~ `@#\$%^&\*()\_+-=[ ]\{}|;':""",./<>?"..

### Description

`IdTrxFrom` is a string of characters that provides the “From” list for DDL Identifier Translation mappings. 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.

For further details, see the “[Identifiers](#)” chapter in *Using InterSystems SQL*. Also see the parameter `IdTrxTo`.

### Changing This Parameter

To set the desired value for `IdTrxFrom` from the InterSystems IRIS® data platform Terminal, use the `SetDDLIdentifierTranslations()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `IdTrxFrom` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## IdTrxTo

Define the “To” list of characters for Identifier Translation.

```
[SQL]      IdTrxTo=n
```

*n* is a string with a maximum length of 256 characters. The default is an empty string.

### Description

`IdTrxTo` is a string of characters that provides the “To” list for the DDL Identifier Translation mappings.

For further details, see the “[Identifiers](#)” chapter in *Using InterSystems SQL*. Also see [IdTrxFrom](#).

### Changing This Parameter

To set the desired value for `IdTrxTo` from the InterSystems IRIS® data platform Terminal, use the `SetDDLIdentifierTranslations()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `IdTrxTo` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# JavaClassPath

Define additional .jar files to load when the JDBC service is started.

```
[SQL] JavaClassPath=jar file, jar file...
```

*jar file* is the full path to a valid .jar file. By default, no files are listed.

## Description

JavaClassPath is a comma-separated list of .jar files.

When JDBC gateway starts, the default .jar files that are loaded are specified in the intersystems-gateway-3.0.0.jar file. If you want additional .jar files to load when JDBC gateway starts, specify them with their full path in the JavaClassPath parameter.

## Examples

```
JavaClassPath=C:/JarFolder/example.jar
```

```
JavaClassPath=C:/JarFolder/example.jar, C:/JarFolder/example2.jar
```

## Changing This Parameter

You can change JavaClassPath with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# JavaHome

Specify the Java version to use with the JDBC Gateway.

```
[SQL]      JavaHome=directory
```

*directory* is the full path to a valid directory. By default, no directory is listed.

## Description

JavaHome specifies the full path of the directory that contains the non-default version of Java that you want to run when you start the JDBC Gateway. If this setting is empty, then your system's default version of Java is used.

## Changing This Parameter

On the **JDBC Gateway Server** page of the Management Portal (**System Administration > Configuration > Connectivity > JDBC Gateway Server**), in the **Java Home Directory** field, enter the Java directory.

You can also change JavaHome with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# JDBCGatewayJVMArgs

Specify optional JVM arguments to use when starting the JDBC Gateway.

```
[SQL] JDBCGatewayJVMArgs= args
```

*args* is a string with a maximum length of 1024 characters. The default is an empty string.

## Description

JDBCGatewayJVMArgs contains optional JVM arguments to include in the command line when starting the JDBC Gateway.

## Example

This entry defines the minimum (-Xms) and maximum (-Xmx) JVM heap size:

```
JDBCGatewayJVMArgs=-Xms64m -Xmx512m
```

## Changing This Parameter

On the **JDBC Gateway Server** page of the Management Portal (**System Administration > Configuration > Connectivity > JDBC Gateway Server**), in the **JVM Arguments** field, enter arguments.

You can also change JDBCGatewayJVMArgs with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# JDBCGatewayLog

Set the name of the JDBC Gateway log file.

```
[SQL] JDBCGatewayLog=n
```

*n* is a string with a maximum length of 256 characters. The default is an empty string.

## Description

JDBCGatewayLog provides the name of the JDBC Gateway log file. There is no default name. You must specify a name if you wish logging to occur.

## Changing This Parameter

On the **JDBC Gateway Server** page of the Management Portal (**System Administration > Configuration > Connectivity > JDBC Gateway Server**), in the **Log File** field, enter the name of the JDBC Gateway log file.

You can also change JDBCGatewayLog with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## JDBCGatewayPort

Set the port number for the JDBC Gateway.

```
[SQL] JDBCGatewayPort=n
```

*n* is a valid port number. The default is 62972.

### Description

`JDBCGatewayPort` is the port number for the JDBC Gateway; a five-digit integer. The default value is 62972. If you have multiple InterSystems IRIS® data platform instances on your system, this port number is incremented for each successive InterSystems IRIS instance.

### Changing This Parameter

On the **JDBC Gateway Server** page of the Management Portal (**System Administration > Configuration > Connectivity > JDBC Gateway Server**), in the **Port** field, enter the number of the JDBC Gateway port.

You can also change `JDBCGatewayPort` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# JDBCGatewayUsePassphrase

Enable the passphrase requirement for JDBC connections.

```
[SQL] JDBCGatewayUsePassphrase=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When `JDBCGatewayUsePassphrase` is enabled (*n* = 1), a passphrase is required for JDBC connection. Otherwise, no passphrase is required.

## Changing This Parameter

On the **JDBC Gateway Server** page of the Management Portal (**System Administration > Configuration > Connectivity > JDBC Gateway Server**), select **Use Passphrase** to require a passphrase.

You can also change `JDBCGatewayUsePassphrase` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# LockThreshold

Set the SQL lock threshold.

```
[SQL]      LockThreshold=n
```

*n* is any nonnegative integer. The default value is 1000.

## Description

The value *n* is the lock threshold. This is the number of 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, after the 1001<sup>st</sup> 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.

For further details, see the [Transaction Locking](#) section of the “INSERT” reference page in *InterSystems SQL Reference*.

## Changing This Parameter

On the **SQL** page of the Management Portal ([System Administration > Configuration > SQL and Object Settings > SQL](#)), in the **Lock escalation threshold** field, enter a number.

To set the desired value for `LockThreshold` from the InterSystems IRIS® data platform Terminal, use the `SetLockThreshold()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `LockThreshold` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# LockTimeout

Set the SQL lock timeout.

```
[SQL]      LockTimeout=n
```

*n* is an integer in the range 0—32,767. The default value is 10.

## Description

`LockTimeout` is the lock timeout (in seconds) for InterSystems IRIS® data platform locks made during execution of SQL statements. The maximum value is 32,767 seconds, or 9 hours.

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

## Changing This Parameter

On the **SQL** page of the Management Portal ([System Administration > Configuration > SQL and Object Settings > SQL](#)), for the **Lock Timeout (seconds)** setting, enter a number.

To set the desired value for `LockTimeout` from the InterSystems IRIS Terminal, use the `SetLockTimeout()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `LockTimeout` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# ODBCVarcharMaxlen

Set the MaxLen for ODBC fields of type VarChar.

```
[SQL]      ODBCVarcharMaxlen=n
```

*n* is any nonnegative integer. The default value is 4096.

## Description

ODBCVarcharMaxlen is the MaxLen (maximum length) that InterSystems IRIS® data platform will report to ODBC for fields with the data type VarChar.

## Changing This Parameter

You can change ODBCVarcharMaxlen with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# QueryProcedures

Specify whether all class queries project as SQL Stored Procedures.

```
[SQL]      QueryProcedures=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When **QueryProcedures** is enabled (*n* = 1), all SQL class queries project as SQL Stored Procedures, regardless of the query's **SqlProc** value. When this parameter is not enabled, only class queries defined with **SqlProc=1** project as Stored Procedures.

When changing this setting, you must recompile the classes with the class queries in order for this change to have an affect. Modifying this setting in the CPF does not require a system restart to make it active.

## Changing This Parameter

To set the desired value for **QueryProcedures** from the InterSystems IRIS® data platform Terminal, use the **SetPARAMETER()** method of the **%SYSTEM.SQL** class. See the class reference for details.

You can also change **QueryProcedures** with the **SetQueryProcedures()** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# RTPC

Enable Runtime Plan Choice (RTPC) query optimization.

```
[SQL] RTPC=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When RTPC is enabled (*n* = 1), InterSystems SQL queries can perform optimization based on [outlier information](#). For further details on outlier selectivity, refer to [Tune Table](#) in the “Optimizing Tables” chapter in *InterSystems SQL Optimization Guide*.

RTPC cannot be set to 1 if the [BiasQueriesAsOutlier](#) parameter is also set to 1. When [BiasQueriesAsOutlier](#) is set, InterSystems SQL is optimized for queries that often refer to the outlier.

## Changing This Parameter

On the **SQL** page of the Management Portal ([System Administration > Configuration > SQL and Object Settings > SQL](#)), for the **Optimize queries based on parameter values** setting, choose Optimize for actual query parameter values at runtime.

To set the desired value for RTPC from the InterSystems IRIS® data platform Terminal, use the **SetRTPC()** method of the **%SYSTEM.SQL** class. See the class reference for details.

You can also change RTPC with the **Config.SQL** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# ReferentialChecks

Enable foreign key constraint validation.

```
[SQL] ReferentialChecks=n
```

*n* is either 1 or 0. The default value is 1.

## Description

When `ReferentialChecks` is enabled (*n* = 1), InterSystems IRIS® data platform validates the foreign key constraint for `INSERT`, `UPDATE`, `DELETE`, and `TRUNCATE TABLE` operations. When this parameter is not enabled, InterSystems IRIS bypasses validation of foreign key constraints.

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

## Changing This Parameter

To set the desired value for `ReferentialChecks` from the InterSystems IRIS Terminal, use the `SetFilerRefIntegrity()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `ReferentialChecks` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# SaveMAC

Save the source code for cached query routines.

```
[SQL]      SaveMAC=n
```

*n* is either 1 or 0. The default value is 0.

## Description

When `SaveMac` is enabled (*n* = 1), the source code (.MAC and .INT) for cached query routines created through Dynamic SQL is saved.

## Changing This Parameter

On the **SQL** page of the Management Portal (**System Administration > Configuration > SQL and Object Settings > SQL**), select **Retain cached query source** to enable `SaveMac`.

To set the desired value for `SaveMac` from the InterSystems IRIS® data platform Terminal, use the `SetCachedQuerySaveSource()` method of the `%SYSTEM.SQL` class. See the class reference for details.

You can also change `SaveMac` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# TCPKeepAlive

Set the number of seconds between keep-alive messages.

```
[SQL]      TCPKeepAlive=n
```

*n* is an integer in the range 30—432,000. The default value is 300.

## Description

TCPKeepAlive is the number of seconds between keep-alive messages. The setting applies only to InterSystems IRIS® data platform running on Windows and Linux. The default is 300 seconds (5 minutes), and the maximum value is 432,000 (5 days). If the value is 0, the system uses the operating system default.

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

## Changing This Parameter

On the **SQL** page of the Management Portal ([System Administration > Configuration > SQL and Object Settings > SQL](#)), enter a value for the **TCP keepalive for client connections (seconds)** setting.

To set the desired value for **PARAMETER** from the InterSystems IRIS Terminal, use the **SetTCPKeepAlive()** method of the **%SYSTEM.SQL** class. See the class reference for details.

You can also change **PARAMETER** with the **Config.SQL** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

For a TCP device, you can also specify this interval with the OPEN or USE commands, as the eighth parameter (for example: `O tcp: ( "SERVER" :port :mode :::::keepalive)`). or the keyword `/KEEPALIVE=n`. The OPEN command parameters for a TCP device are:

`hostname{ :port{ :mode{ :terminators{ :ibfsz{ :obfsz{ :queuesize{ :keepalivetime} } } } } }`. For details about the **OPEN** and **USE** commands and arguments, including examples, see the [ObjectScript Reference](#).

# TimePrecision

Set the default time precision for SQL scalar time functions.

```
[SQL]      TimePrecision=n
```

*n* is an integer in the range 0—9. The default value is 0.

## Description

TimePrecision defines the default number of decimal places in the value returned by the SQL scalar functions **GETDATE()**, **CURRENT\_TIME**, **CURRENT\_TIMESTAMP**, **GETUTCDATE**, and **UNIX\_TIMESTAMP**. A value returned by these function has *n* decimal places for fractional seconds. The actual precision possible is platform dependant; precision digits in excess of the precision available on your system are returned as zeros.

## Changing This Parameter

On the **SQL** page of the Management Portal (**System Administration > Configuration > SQL and Object Settings > SQL**), for the **Default time precision for GETDATE(), CURRENT\_TIME, and CURRENT\_TIMESTAMP** setting, select a number of decimal places.

To set the desired value for TimePrecision from the InterSystems IRIS® data platform Terminal, use the **SetDefaultTimePrecision()** method of the **%SYSTEM.SQL** class. See the class reference for details.

You can also change TimePrecision with the **Config.SQL** class (as described in the class reference), by editing the CPF in a text editor (as described in the **Editing the Active CPF** section of the “Introduction to the Configuration Parameter File” chapter in this book), or by using the **SET OPTION** command.

# TODATEDefaultFormat

Set the default date format for the SQL TO\_DATE() function.

```
[SQL]      TODATEDefaultFormat=n
```

*n* is any string in a format appropriate for the **TO\_DATE()** function. The default string is DD MON YYYY.

## Description

The value *n* provides the format string that the SQL TO\_DATE() function uses when TO\_DATE() is called without a format specified.

## Examples

The following is an example:

```
TODATEDefaultFormat=DD MON YYYY
```

And the following another example:

```
TODATEDefaultFormat=YYYY DD MM
```

For more examples and an in-depth discussion about valid date strings, see the [Format](#) section of the “TO\_DATE” reference page in *InterSystems RSQL Reference*.

## Changing This Parameter

On the **SQL** page of the Management Portal (**System Administration > Configuration > SQL and Object Settings > SQL**), in the **TO\_DATE default format** field, enter a valid format string.

To set the desired value for TODATEDefaultFormat from the InterSystems IRIS® data platform Terminal, use the **SetToDateDefaultFormat()** method of the %SYSTEM.SQL class. See the class reference for details.

You can also change TODATEDefaultFormat with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).



# [SqlSysDatatypes]

This topic describes the parameters found in the [SqlSysDatatypes] section of the CPF.

# System Datatypes

Map SQL datatypes to their InterSystems IRIS® data platform equivalents.

```
[SqlSysDatatypes]      x=a
```

*x* is the name of an SQL datatype. *a* is the InterSystems IRIS equivalent.

## Description

The [SqlSysDatatypes] section contains system-defined datatype descriptions. Each description maps an SQL datatype to its InterSystems IRIS equivalent in the format *x=a*, as follows:

- Each keyword *x* is the name of the SQL datatype, plus any allowed arguments.
- The value *a* is the InterSystems IRIS equivalent, including any constraints on the arguments.

## Example

The following excerpt of a configuration parameter file shows the [SqlSysDatatypes] which begin with the letter **B**. For a table of all System-Defined DDL Datatype mappings, see [Table of DDL Data Types](#) in the “Data Types” chapter of *InterSystems SQL Reference*.

```
[SqlSysDatatypes]
BIGINT=%Library.BigInt
BIGINT(%1)=%Library.BigInt
BINARY=%Library.Binary(MAXLEN=1)
BINARY VARYING=%Library.Binary(MAXLEN=1)
BINARY VARYING(%1)=%Library.Binary(MAXLEN=%1)
BINARY(%1)=%Library.Binary(MAXLEN=%1)
BIT=%Library.Boolean
```

## Changing This Parameter

On the **System-defined DDL Mappings** page of the Management Portal ([System Administration > Configuration > SQL and Object Settings > System DDL Mappings](#)), select **Edit** to modify a datatype definition.

Instead of using the Management Portal, you can modify datatype definitions by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

To create your own datatype, see the [User Datatypes](#) reference in this book.

For more information about SQL datatypes, see the [Data Types](#) chapter in *InterSystems SQL Reference*.

# [SqlUserDatatypes]

This topic describes the parameters found in the [SqlUserDatatypes] section of the CPF.

# User Datatypes

Map SQL datatypes to their InterSystems IRIS® data platform equivalents.

```
[SqlUserDatatypes]      x=a
```

*x* is the name of an SQL datatype. *a* is the InterSystems IRIS equivalent.

## Description

The [SqlUserDatatypes] section contains a user-defined datatype descriptions. Each description maps an SQL datatype to its InterSystems IRIS equivalent in the format *x=a*, as follows:

- Each keyword *x* is the name of the SQL datatype, plus any allowed arguments.
- The value *a* is the InterSystems IRIS equivalent, including any constraints on the arguments.

## Example

With the following line in the CPF, when MYVARCHAR(10) is seen in a statement, the property is created with type %Library.String(MAXLEN=10,TRUNCATE=0).

```
[SqlUserDatatypes]
MYVARCHAR(%1)=%Library.String(MAXLEN=%1,TRUNCATE=0)
```

## Changing This Parameter

On the **User-defined DDL Mappings** page of the Management Portal (**System Administration > Configuration > SQL and Object Settings > User DDL Mappings**), click **Create New User-defined DDL Mapping** to add a new datatype mapping, or click **Edit** to modify an existing one.

Instead of using the Management Portal, you can modify datatype definitions in the Config.SqlUserDatatypes class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

To review system-defined datatypes, see the [System Datatypes](#) reference in this book.

For more information about SQL datatypes, see the [Data Types](#) chapter in *InterSystems SQL Reference*.

# [Startup]

This topic describes the parameters found in the [Startup] section of the CPF.

## CallinHalt

Allow custom routines during callin close.

```
[Startup]    CallinHalt=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

### Description

When **CallinHalt** is enabled (*n* = 1), InterSystems IRIS® data platform executes the CALLIN^%ZSTOP routine entry each time an external program ends a CALLIN. When this parameter is not enabled, the routine is not executed.

For more information about CALLIN, see *Using The Callin API*.

### Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **CallinHalt** row, select **Edit**. Select **CallinHalt** to enable this setting.

Instead of using the Management Portal, you can change **CallinHalt** in the Config.Startup class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

### See Also

[ProcessStart](#) and [ProcessHalt](#) correspond to *foreground* processes. These are processes that are started via a terminal session or are specifically set to run in the foreground.

[JobStart](#) and [JobHalt](#) correspond to *background* processes. This includes any processes that are started via the JOB command, plus any background server processes including ODBC or any of the object bindings.

[SystemStart](#) and [SystemHalt](#) correspond to InterSystems IRIS system startup or shutdown.

[CallinStart](#) and [CallinHalt](#) correspond to external programs performing a CALLIN.

For more information about all these parameters, see [Customizing Start and Stop Behavior with ^%ZSTART and ^%ZSTOP](#) in the “Customizing the InterSystem IRIS System” chapter in *Specialized System Tools and Utilities*.

# CallinStart

Allow custom routines during callin initialization.

```
[Startup] CallinStart=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When **CallinStart** is enabled (*n* = 1), InterSystems IRIS® data platform executes the CALLIN^%ZSTART routine entry each time an external program begins a CALLIN. When this parameter is not enabled, the routine is not executed.

For more information about CALLIN, see *Using The Callin API*.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **CallinStart** row, select **Edit**. Select **CallinStart** to enable this setting.

Instead of using the Management Portal, you can change **CallinStart** in the Config.Startup class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

[ProcessStart](#) and [ProcessHalt](#) correspond to *foreground* processes. These are processes that are started via a terminal session or are specifically set to run in the foreground.

[JobStart](#) and [JobHalt](#) correspond to *background* processes. This includes any processes that are started via the JOB command, plus any background server processes including ODBC or any of the object bindings.

[SystemStart](#) and [SystemHalt](#) correspond to InterSystems IRIS system startup or shutdown.

[CallinStart](#) and [CallinHalt](#) correspond to external programs performing a CALLIN.

For more information about all these parameters, see [Customizing Start and Stop Behavior with ^%ZSTART and ^%ZSTOP](#) in the “Customizing the InterSystem IRIS System” chapter in *Specialized System Tools and Utilities*.

## CliSysName

Set the node name for the local system.

```
[Startup]      Name=n
```

*n* is a string with a maximum length of 64 characters. The default is an empty string.

### Description

CliSysName is the node name for this computer, and is used as:

- 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.
- The node name returned by certain forms of the \$SYSTEM function, concatenated with the InterSystems IRIS® data platform instance name, as *nodename:instancename*. This concatenated string is recorded in Audit files.

If no name is provided, InterSystems IRIS reads the computer settings and uses the computer “host name” as the client node name.

### Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **CliSysName** row, select **Edit**. Enter the desired node name.

Instead of using the Management Portal, you can change CliSysName in the Config.Startup class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DBSizesAllowed

Specify possible database block sizes when creating a database.

```
[Startup]    DBSizesAllowed=n[n,n...]
```

*n* can be 8192, 16384, 32768, or 65536. The default value is 8192.

## Description

DBSizesAllowed lists the database block sizes (in bytes) that you can select when creating a database.

For more information about creating and managing databases, see the [Configuring Databases](#) section of the “Configuring InterSystems IRIS” chapter in *System Administration Guide*.

## Examples

```
DBSizesAllowed=8192,16384
```

```
DBSizesAllowed=8192,65536
```

## Changing This Parameter

On the **Startup** page of the Management Portal ([System Administration > Configuration > Additional Settings > Startup](#)), in the **DBSizesAllowed** row, select **Edit**. Select each desired database block size. You cannot clear 8192.

**Important:** When you enable an additional database block size, you *must* allocate memory for that block size using the [globals](#) parameter. This allows InterSystems IRIS® data platform to create the needed pool of global buffers for that size.

Instead of using the Management Portal, you can change DBSizesAllowed in the Config.Startup class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF file](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

- [globals](#) parameter in the [Config] section of this reference.
- [Configuring Databases](#) in the “Configuring InterSystems IRIS” chapter in *System Administration Guide*.
- [Allocating Memory to the Routine and Database Caches](#) in the “Configuring InterSystems IRIS” chapter in *System Administration Guide*.
- [Large Block Size Considerations](#) in the “Configuring InterSystems IRIS” chapter in *System Administration Guide*.

## DefaultPort

Set the port number for the InterSystems IRIS® data platform superserver.

```
[Startup] DefaultPort=n
```

*n* is a valid port number. The default is 51773.

### Description

`DefaultPort` is the port number for the InterSystems IRIS superserver. The superserver listens on a specified port (51773 by default) for incoming connections to InterSystems IRIS and dispatches them to the appropriate subsystem.

A standard InterSystems IRIS installation sets the superserver port number to 51773, or if that port is in use by another InterSystems IRIS instance on the same system, the next unused port number.

### Changing This Parameter

On the **Memory and Startup** page of the Management Portal (**System Administration > Configuration > System Configuration > Memory and Startup**), enter a number in the **Superserver Port Number** field.

Instead of using the Management Portal, you can change `DefaultPort` in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# DefaultPortBindAddress

Set the IP address for the InterSystems IRIS® data platform superserver to bind to.

```
[Startup] DefaultPortBindAddress=nnn.nnn.nn.nn
```

*nnn.nnn.nn.nn* is a valid IP address. By default, none is specified.

## Description

**DefaultPortBindAddress** is the IP addresses on the host system that the superserver should bind to. The superserver is the process that accepts client connections for ODBC, JDBC, and other connection technologies. 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.

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.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **DefaultPortBindAddress** row, select **Edit**. Enter an IP address.

Instead of using the Management Portal, you can change **DefaultPortBindAddress** in the **Config.Startup** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# EnableVSSBackup

Allow Volume Shadow Copy Service (VSS) backup. Windows systems only.

```
[Startup]    EnableVSSBackup=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When `EnableVSSBackup` is enabled (*n* = 1), InterSystems IRIS® data platform supports VSS on Windows. VSS is only available on Windows. On other platforms, InterSystems IRIS ignores the `EnableVSSBackup` parameter.

See the [Backup Strategies](#) section in the “Backup and Restore” chapter in *Data Integrity Guide* for information about creating a backup using VSS or other methods.

## Changing This Parameter

On the **Startup** page of the Management Portal ([System Administration > Configuration > Additional Settings > Startup](#)), in the `EnableVSSBackup` row, select **Edit**. Select `EnableVSSBackup` to enable this setting.

Instead of using the Management Portal, you can change `EnableVSSBackup` in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# EnsembleAutoStart

Allow productions to auto-start when InterSystems IRIS® data platform starts.

```
[Startup] EnsembleAutoStart=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When **EnsembleAutoStart** is enabled, the production you set to auto-start in each interoperability-enabled namespace starts when you start InterSystems IRIS. To facilitate debugging situations involving troubled productions, you can disable this setting to prevent a production from starting.

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

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **EnsembleAutoStart** row, select **Edit**. Select **EnsembleAutoStart** to enable this setting.

Instead of using the Management Portal, you can change **EnsembleAutoStart** in the **Config.Startup** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# ErrorPurge

Set the number of days to keep error globals.

```
[Startup]    ErrorPurge=n
```

*n* is an integer in the range 1—1000. The default value is 30.

## Description

`ErrorPurge` is the number of days to keep the error globals for the `^%ETN` error handler. Errors older than this are deleted on the next InterSystems IRIS® data platform restart.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **ErrorPurge** row, select **Edit**. Enter a number of days.

Instead of using the Management Portal, you can change `ErrorPurge` in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

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

# FIPSMode

Enable FIPS 140–2 compliant library for database encryption on Red Hat Linux.

```
[Startup]    FIPSMode=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

When **FIPSMode** is enabled, InterSystems IRIS® data platform 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.

See the article [FIPS 140–2 Compliance for Database Encryption](#) for details.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **FIPSMode** row, select **Edit**. Select **FIPSMode** to use the FIPS 140-2 compliant library for database encryption.

Instead of using the Management Portal, you can change **FIPSMode** in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# IPv6

Allow InterSystems IRIS® data platform to accept IPv6 addresses.

```
[Startup]    IPv6=n
```

*n* is either 1 (true) or 0 (false). The default value is 0.

## Description

`IPv6` controls whether your system is operating in an IPv6 (Internet Protocol Version 6) network, with IPv6 addresses. For more information, see the [IPv6 Support](#) section in the “Configuring InterSystems IRIS” chapter in *System Administration Guide*.

## Changing This Parameter

On the **Startup** page of the Management Portal ([System Administration > Configuration > Additional Settings > Startup](#)), in the **IPv6** row, select **Edit**. Select **IPv6** to enable this setting.

Instead of using the Management Portal, you can change `IPv6` in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# JobHalt

Allow custom routines during background process (job) shutdown.

```
[Startup] JobHalt=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When `JobHalt` is enabled (*n* = 1), InterSystems IRIS® data platform executes the `JOB^%ZSTOP` routine entry when a background process ends. Background processes include any processes that are started via the `JOB` command, plus any background server processes including ODBC or any of the language bindings. When this parameter is not enabled, the routine is not executed.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **JobHalt** row, select **Edit**. Select **JobHalt** to enable this setting.

Instead of using the Management Portal, you can change `JobHalt` in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

[ProcessStart](#) and [ProcessHalt](#) correspond to *foreground* processes. These are processes that are started via a terminal session or are specifically set to run in the foreground.

[JobStart](#) and `JobHalt` correspond to *background* processes. This includes any processes that are started via the `JOB` command, plus any background server processes including ODBC or any of the object bindings.

[SystemStart](#) and [SystemHalt](#) correspond to InterSystems IRIS system startup or shutdown.

[CallinStart](#) and [CallinHalt](#) correspond to external programs performing a CALLIN.

For more information about all these parameters, see [Customizing Start and Stop Behavior with ^%ZSTART and ^%ZSTOP](#) in the “Customizing the InterSystem IRIS System” chapter in *Specialized System Tools and Utilities*.

# JobServers

Set the number of job servers.

```
[Startup]    JobServers=n
```

*n* is an integer in the range 0—2000. The default value is 0.

## Description

**JobServers** is the number of job servers you want InterSystems IRIS® data platform to start up.

Having a large number of job servers running will use more memory and processes, but allows for much faster jobbing of processes because InterSystems IRIS 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.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **JobServers** row, select **Edit**. Enter a number of job servers.

Instead of using the Management Portal, you can change **JobServers** in the **Config.Startup** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# JobStart

Allow custom routines during background process (job) startup.

```
[Startup] JobStart=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When `JobStart` is enabled (*n* = 1), InterSystems IRIS® data platform executes the `JOB^%ZSTART` routine entry when a background process starts. Background processes include any processes that are started via the `JOB` command, plus any background server processes including ODBC or any of the language bindings. When this parameter is not enabled, the routine is not executed.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **JobStart** row, select **Edit**. Select **JobStart** to enable this setting.

Instead of using the Management Portal, you can change `JobStart` in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

[ProcessStart](#) and [ProcessHalt](#) correspond to *foreground* processes. These are processes that are started via a terminal session or are specifically set to run in the foreground.

`JobStart` and [JobHalt](#) correspond to *background* processes. This includes any processes that are started via the `JOB` command, plus any background server processes including ODBC or any of the object bindings.

[SystemStart](#) and [SystemHalt](#) correspond to InterSystems IRIS system startup or shutdown.

[CallinStart](#) and [CallinHalt](#) correspond to external programs performing a `CALLIN`.

For more information about all these parameters, see [Customizing Start and Stop Behavior with ^%ZSTART and ^%ZSTOP](#) in the “Customizing the InterSystem IRIS System” chapter in *Specialized System Tools and Utilities*.

## LicenseID

Allow instance to requests a key from the license server.

```
[Startup]    LicenseID=n
```

*n* is the name of a license key in the target .key file. By default, no key is specified.

### Description

If there is no local iris.key file, InterSystems IRIS® data platform uses LicenseID to request a license key from the License Server. Each license key loaded in the License Server will have a unique LicenseID.

For more information about license keys, see the [Managing InterSystems IRIS Licensing](#) chapter in *System Administration Guide*.

### Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **LicenseID** row, select **Edit**. Enter a LicenseID.

Instead of using the Management Portal, you can change LicenseID in the Config.Startup class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# MaxConsoleLogSize

Set the maximum size of the messages.log file.

```
[Startup]      MaxConsoleLogSize=n
```

*n* is an integer in the range 1—500 (MB). The default value is 5.

## Description

MaxConsoleLogSize is the maximum size of the InterSystems IRIS® data platform console file, in megabytes. The console file is `messages.log`, located in the `install-dir\mgr` directory by default, or the directory specified by the `console` parameter.

If you enter a value that is smaller than the current setting of MaxConsoleLogSize, or if the console file grows to reach the size limit, then the current `messages.log` file is renamed to `messages.old_Date`. The system creates an empty `messages.log` file, and new entries are appended to the newly-created file.

You can view the messages log on **Messages Log** page of the Management Portal (**System Operation > System Logs > Messages Log**). To configure the location of the `messages.log` file, see `ConsoleFile` parameter.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **MaxConsoleLogSize** row, select **Edit**. Enter a number of megabytes.

Instead of using the Management Portal, you can change MaxConsoleLogSize in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in the **Editing the Active CPF** section of the “Introduction to the Configuration Parameter File” chapter in this book).

# MaxIRISTempSizeAtStart

Set the maximum size of the IRISTEMP database on restart.

```
[Startup]    MaxIRISTempSizeAtStart=n
```

*n* is an integer in the range 0—1,000,000 (MB). The default value is 0.

## Description

**MaxIRISTempSizeAtStart** is the maximum size in megabytes of the IRISTEMP database when the system is restarted. When the system restarts, the IRISTEMP database is truncated to this size. If 0, the IRISTEMP database is not truncated.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **MaxIRISTempSizeAtStart** row, select **Edit**. Enter a number of megabytes.

Instead of using the Management Portal, you can change **MaxIRISTempSizeAtStart** in the **Config.Startup** class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# ProcessHalt

Allow custom routines during foreground process shutdown.

```
[Startup]    ProcessHalt=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When `ProcessHalt` is enabled (*n* = 1), InterSystems IRIS® data platform executes the `LOGIN^%ZSTOP` routine entry at foreground process logout (such as when a user closes the terminal). When this parameter is not enabled, the routine is not executed.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **ProcessHalt** row, select **Edit**. Select **ProcessHalt** to enable this setting.

Instead of using the Management Portal, you can change `ProcessHalt` in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

[ProcessStart](#) and [ProcessHalt](#) correspond to *foreground* processes. These are processes that are started via a terminal session or are specifically set to run in the foreground.

[JobStart](#) and [JobHalt](#) correspond to *background* processes. This includes any processes that are started via the `JOB` command, plus any background server processes including ODBC or any of the object bindings.

[SystemStart](#) and [SystemHalt](#) correspond to InterSystems IRIS system startup or shutdown.

[CallinStart](#) and [CallinHalt](#) correspond to external programs performing a `CALLIN`.

For more information about all these parameters, see [Customizing Start and Stop Behavior with ^%ZSTART and ^%ZSTOP](#) in the “Customizing the InterSystem IRIS System” chapter in *Specialized System Tools and Utilities*.

# ProcessStart

Allow custom routines during foreground process startup.

```
[Startup]    ProcessStart=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When `ProcessStart` is enabled (*n* = 1), InterSystems IRIS® data platform executes the `LOGIN^%ZSTART` routine entry at foreground process login (such as when a user logs in to the terminal). When this parameter is not enabled, the routine is not executed.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **ProcessStart** row, select **Edit**. Select **ProcessStart** to enable this setting.

Instead of using the Management Portal, you can change `ProcessStart` in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

`ProcessStart` and `ProcessHalt` correspond to *foreground* processes. These are processes that are started via a terminal session or are specifically set to run in the foreground.

`JobStart` and `JobHalt` correspond to *background* processes. This includes any processes that are started via the `JOB` command, plus any background server processes including ODBC or any of the object bindings.

`SystemStart` and `SystemHalt` correspond to InterSystems IRIS system startup or shutdown.

`CallinStart` and `CallinHalt` correspond to external programs performing a CALLIN.

For more information about all these parameters, see [Customizing Start and Stop Behavior with ^%ZSTART and ^%ZSTOP](#) in the “Customizing the InterSystem IRIS System” chapter in *Specialized System Tools and Utilities*.

# ShutdownTimeout

Set the number of seconds InterSystems IRIS® data platform should wait until forcing a shutdown.

```
[Startup]     ShutdownTimeout=n
```

*n* is an integer in the range 120—100,000. The default value is 300 (5 minutes).

## Description

ShutdownTimeout is the number of seconds InterSystems IRIS should wait for shutdown to complete normally before timing out and forcing a shutdown.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **ShutdownTimeout** row, select **Edit**. Enter a number of seconds.

Instead of using the Management Portal, you can change ShutdownTimeout in the Config.Startup class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# SystemHalt

Allow custom routines during system shutdown.

```
[Startup] SystemHalt=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When **SystemHalt** is enabled (*n* = 1), InterSystems IRIS® data platform executes the SYSTEM^%ZSTOP routine entry at system shutdown. When this parameter is not enabled, the routine is not executed.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **SystemHalt** row, select **Edit**. Select **SystemHalt** to enable this setting.

Instead of using the Management Portal, you can change **SystemHalt** in the Config.Startup class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

[ProcessStart](#) and [ProcessHalt](#) correspond to *foreground* processes. These are processes that are started via a terminal session or are specifically set to run in the foreground.

[JobStart](#) and [JobHalt](#) correspond to *background* processes. This includes any processes that are started via the JOB command, plus any background server processes including ODBC or any of the object bindings.

[SystemStart](#) and [SystemHalt](#) correspond to InterSystems IRIS system startup or shutdown.

[CallinStart](#) and [CallinHalt](#) correspond to external programs performing a CALLIN.

For more information about all these parameters, see [Customizing Start and Stop Behavior with ^%ZSTART and ^%ZSTOP](#) in the “Customizing the InterSystem IRIS System” chapter in *Specialized System Tools and Utilities*.

# SystemStart

Allow custom routines during system startup.

```
[Startup] SystemStart=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When **SystemStart** is enabled (*n* = 1), InterSystems IRIS® data platform executes the SYSTEM^%ZSTART routine entry at system startup. When this parameter is not enabled, the routine is not executed.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **SystemStart** row, select **Edit**. Select **SystemStart** to enable this setting.

Instead of using the Management Portal, you can change **SystemStart** in the Config.Startup class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

[ProcessStart](#) and [ProcessHalt](#) correspond to *foreground* processes. These are processes that are started via a terminal session or are specifically set to run in the foreground.

[JobStart](#) and [JobHalt](#) correspond to *background* processes. This includes any processes that are started via the JOB command, plus any background server processes including ODBC or any of the object bindings.

[SystemStart](#) and [SystemHalt](#) correspond to InterSystems IRIS system startup or shutdown.

[CallinStart](#) and [CallinHalt](#) correspond to external programs performing a CALLIN.

For more information about all these parameters, see [Customizing Start and Stop Behavior with ^%ZSTART and ^%ZSTOP](#) in the “Customizing the InterSystem IRIS System” chapter in *Specialized System Tools and Utilities*.

# TempDirectory

Specify the subdirectory for temporary files.

```
[Startup] TempDirectory=n
```

*n* is an absolute or relative directory pathname. The default is Temp, which corresponds to <install-dir>\mgr\Temp.

## Description

TempDirectory is the name of the subdirectory for InterSystems IRIS® data platform to store temporary files. When you set a new TempDirectory value, the system creates a subdirectory of this name which becomes the new InterSystems IRIS temporary directory.

You can specify a full or relative path. If you specify a full path, InterSystems IRIS uses the specified directory. If you specify a relative path, InterSystems IRIS creates the directory under the <install-dir>\mgr\ subdirectory.

## Examples

To create c:\InterSystems\iris\mgr\Temp\ on Windows:

```
TempDirectory=Temp
```

To create c:\TempFiles\ on Windows:

```
TempDirectory=c:\TempFiles
```

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **TempDirectory** row, select **Edit**. Enter a subdirectory name.

Instead of using the Management Portal, you can change TempDirectory in the Config.Startup class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# TerminalPrompt

Define the format of the terminal prompt.

```
[Startup] TerminalPrompt=n
```

## Description

`TerminalPrompt` is 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 determines the order the values appear in the prompt. For example

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

gives you a terminal prompt of `%SYS:HostName>`

- 0 - Simple prompt. Specify 0 with no other values to 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 `$NAMESPACE` special variable. It can be an explicit namespace name or an implied namespace name.
- 3 - Config name. The name of your system installation. For example, `IRIS2>`. This is the same for all of your terminal processes.
- 4 - Current time, expressed as local time in 24-hour format with whole seconds. For example, `15:59:36>`. This is the static time value for when the prompt was returned. This value changes for each prompt.
- 5 - The Process ID for your terminal. For example, `2336>`. This is different for each terminal process. This value can also be returned from the `$JOB` special variable.
- 6 - Username. For example, `fred>`. This is the same for all of your terminal processes.
- 7 - Time elapsed 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>`.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **TerminalPrompt** row, select **Edit**. Enter a comma-separated string of values, or 0.

Instead of using the Management Portal, you can change `TerminalPrompt` in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# WebServer

Allow the web server to start.

```
[Startup]    WebServer=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When Webserver is enabled (*n* = 1), the web server starts when InterSystems IRIS® data platform starts. The web server is necessary to enable REST and SOAP APIs, as well as the Management Portal. For information on the private web server, see the section [Minimal Apache Web Server or Private Web Server](#) in the *Web Gateway Configuration Guide*.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **WebServer** row, select **Edit**. Select **WebServer** to enable this setting.

Instead of using the Management Portal, you can change WebServer in the Config.Startup class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

- [WebServerName](#)
- [WebServerPort](#)
- [WebServerURLPrefix](#)

# WebServerName

Set an IP address or a DNS name for the Web server.

```
[Startup]    WebServerName=nnn.nnn.nn.nn (ip address) or www.DNSname.com
```

*nnn.nnn.nn.nn* is a valid IP address. *www.DNSname.com* is a valid DNS name. Only specify one value.

## Description

WebServerName specifies the DNS name or the IP address of the web server that is configured for use with InterSystems IRIS® data platform tools. The web server is necessary to enable REST and SOAP APIs, as well as the Management Portal.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **WebServerName** row, select **Edit**. Enter a DNS name or an IP address.

Instead of using the Management Portal, you can change WebServerName in the Config.Startup class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

- [WebServer](#)
- [WebServerPort](#)
- [WebServerURLPrefix](#)

# WebServerPort

Set the web server port.

```
[Startup]    WebServerPort=nnnnn
```

*n* is a valid port number. The default is 52773.

## Description

**WebServerPort** is the port number to use for the Web server. If you set this number to 0, the Web Server is disabled. A standard InterSystems IRIS® data platform installation sets the Web server port number to the first unused port number greater than or equal to 52773. The Web server runs on the same system as the InterSystems IRIS server.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **WebServerPort** row, select **Edit**. Enter a port number.

Instead of using the Management Portal, you can change **WebServerPort** in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

- [WebServer](#)
- [WebServerName](#)
- [WebServerURLPrefix](#)

# WebServerURLPrefix

Identify the InterSystems IRIS® data platform instance name for the web server to access.

```
[Startup]    WebServerURLPrefix=n
```

*n* is an alphanumeric string. The default is an empty string.

## Description

`WebServerURLPrefix` is used by Studio when constructing URLs. This should match the [CSP Server Instance](#) setting.

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

## Changing This Parameter

On the **Startup** page of the Management Portal ([System Administration > Configuration > Additional Settings > Startup](#)), in the **WebServerURLPrefix** row, select **Edit**. Enter an InterSystems IRIS instance name.

Instead of using the Management Portal, you can change `WebServerURLPrefix` in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

## See Also

- [WebServer](#)
- [WebServerName](#)
- [WebServerPort](#)

# ZSTU

Allow the user-defined startup to run.

```
[Startup]      ZSTU=n
```

*n* is either 1 (true) or 0 (false). The default value is 1.

## Description

When ZSTU is enabled (*n* = 1), InterSystems IRIS® data platform runs the user-defined startup from the ^ZSTU routine.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**), in the **ZSTU** row, select **Edit**. Select **ZSTU** to enable this setting.

Instead of using the Management Portal, you can change ZSTU in the Config.Startup class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

# [Telnet]

This topic describes the parameters found in the [Telnet] section of the configuration parameter file. This section is valid for windows systems only, and ignored on UNIX® systems.

## DNSLookup

Allow DNS lookup of the Telnet client address. Windows systems only.

```
[Telnet] DNSLookup=n
```

*n* is either the string ON or OFF. The default value is ON.

### Description

DNSLookup enables or disables DNS lookup of the client address in the telnet daemon before passing the address to the InterSystems IRIS® data platform process that was created to service the connection. This determines the format of the client address returned by \$IO and \$ZIO in the InterSystems IRIS process.

When DNSLookup is enabled, a DNS lookup of the client address is performed, and the client name is passed to InterSystems IRIS. When DNSLookup is not enabled, no DNS lookup is performed, and the client address is provided in either dotted decimal format (if the connection was via IPV4) or in the colon separated hexadecimal format (if the connection was via IPV6). You should disable this parameter if a DNS server is not available to do the lookup, because a long delay will occur during login if the DNS server is not available.

InterSystems IRIS Telnet settings apply only to Windows configurations in which InterSystems supplies the Telnet servers. This parameter is ignored for UNIX® systems.

### Changing This Parameter

On the **Telnet Settings** page of the Management Portal (**System Administration > Configuration > Device Settings > Telnet Settings**), in the **DNS Lookup** field, choose **ON** or **OFF**.

Instead of using the Management Portal, you can change DNSLookup in the Config.Telnet class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).

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

# Port

Set the Telnet port number. Windows systems only.

```
[Telnet]      Port=n
```

*n* is a valid TCP/IP port number. The default is value 23.

## Description

Port is the TCP/IP port number for Telnet connections. If multiple InterSystems IRIS® data platform configurations are to run on the same host at the same time, a different Telnet port number must be specified for each 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.

InterSystems IRIS Telnet settings apply only to Windows configurations in which InterSystems supplies the Telnet servers. This parameter is ignored for UNIX® systems.

## Changing This Parameter

On the **Telnet Settings** page of the Management Portal (**System Administration > Configuration > Device Settings > Telnet Settings**), in the **Telnet Port Number** field, enter a TCP/IP port number.

Instead of using the Management Portal, you can change Port in the Config.Telnet class (as described in the class reference) or by editing the CPF in a text editor (as described in the [Editing the Active CPF](#) section of the “Introduction to the Configuration Parameter File” chapter in this book).



# A

## CPF Settings in the Management Portal

This appendix contains information about the Management Portal equivalents of parameters in the CPF. The **Configuration** section of the Management Portal (**System Administration > Configuration**), which the structure of this appendix mimics, contains the following submenus:

- [SQL and Object Settings](#)
- [National Language Settings](#)
- [Device Settings](#)
- [Additional Settings](#)

For information on the other pages in the Management Portal, see the [Management Portal Page Reference](#) section of the “Using the Management Portal” chapter in *System Administration Guide*.

### A.1 SQL and Object Settings

The **SQL and Object Settings** menu (**System Administration > Configuration > SQL and Object Settings**) contains pages where you can configure InterSystems SQL settings. These pages are:

- [SQL](#)
- [Objects](#)
- [TSQL Compatibility](#)
- [ISQL Compatibility](#)
- [System DDL Mapping](#)
- [User DDL Mapping](#)

#### A.1.1 SQL

InterSystems IRIS® data platform includes support for many SQL configuration settings. The **SQL** page (**System Administration > Configuration > SQL and Object Settings > SQL**) contains the following tabs:

- [SQL Tab](#)
- [SQL Shell](#)

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

### A.1.1.1 SQL Tab

The **SQL Tab** allows you to configure various SQL settings. The following table contains a brief description of each option, and a link to the corresponding parameter in the CPF.

SQL Tab Setting	Description	Equivalent CPF parameter
<b>Retain cached query source</b>	Save the source code for cached query routines.	<a href="#">SaveMac</a>
<b>Default time precision for GETDATE(), CURRENT_TIME, and CURRENT_TIMESTAMP</b>	Set the default time precision for SQL scalar time functions.	<a href="#">TimePrecision</a>
<b>Lock escalation threshold</b>	Set the SQL lock threshold.	<a href="#">LockThreshold</a>
<b>TO_DATE default format</b>	Set the default date format for the SQL TO_DATE() function.	<a href="#">TODATEDefaultFormat</a>
<b>Default length for VARCHAR</b>	Set the MaxLen for ODBC fields of type VarChar.	<a href="#">ODBC VarcharMaxlen</a>
<b>Default schema</b>	Set the default SQL schema name.	<a href="#">DefaultSchema</a>
<b>Execute queries in a single process</b>	Allow parallel processing system-wide.	<a href="#">AutoParallel</a>
<b>Define primary key as ID key for tables created via DDL</b>	Set primary key constraint behavior.	<a href="#">IdKey</a>
<b>Ignore redundant DDL statements</b>	Suppress certain SQL errors.	All the <code>DDLNO*</code> parameters in the <a href="#">[SQL]</a> section.
<b>GROUP BY and DISTINCT queries must produce original values</b>	Disallow SQL DISTINCT optimization.	<a href="#">FastDistinct</a>
<b>Optimize queries based on parameter values</b>	Configure SQL query optimization.	<a href="#">BiasQueriesAsOutlier</a> <a href="#">RTPC</a>
<b>Lock timeout (seconds)</b>	Set the SQL lock timeout.	<a href="#">[SQL] LockTimeout</a>

SQL Tab Setting	Description	Equivalent CPF parameter
TCP keepalive for client connections	Set the number of seconds between keep-alive messages.	[SQL] TCPKeepAlive

### A.1.1.2 SQL Shell Tab

The **SQL Shell Tab** allows you to configure the SQL shell. You can use the SQL shell to execute SQL statements dynamically. For more information, see the [Configuring the SQL Shell](#) section of the “Using the SQL Shell Interface” chapter in *Using InterSystems SQL*.

## A.1.2 Objects

The **General Object Settings** page (**System Administration > Configuration > SQL and Object Settings > Objects**) contains a swizzling option. For more information, see [Swizzling a Fieldname Property with %ObjectSelectMode=1](#) in the “Using Dynamic SQL” chapter in *Using InterSystems SQL*.

## A.1.3 TSQL Compatibility

From the **TSQL Settings** page (**System Administration > Configuration > SQL and Object Settings > TSQL Compatibility**), you can configure the behavior of the compiler and colorizer. See the [TSQL Settings](#) chapter in *Transact-SQL (TSQL) Migration Guide* for more information.

## A.1.4 ISQL Compatibility

From the **Informix SQL Settings** page (**System Administration > Configuration > SQL and Object Settings > ISQL Compatibility**), you can configure settings to aid in the migration of ISQL to InterSystems SQL. See the [ISQL Configuration Settings](#) section of the “Overview” chapter in *ISQL Migration Guide* for more information.

## A.1.5 System DDL Mappings

The **System-Defined DDL Mappings** page (**System Administration > Configuration > SQL and Object Settings > System DDL Mappings**) displays system mappings of SQL datatypes to their InterSystems IRIS equivalents. See the [System Datatypes](#) reference for more information about this section. For more information about SQL datatypes, see the [Data Types](#) chapter in *InterSystems SQL Reference*.

## A.1.6 User DDL Mappings

From the **User-Defined DDL Mappings** page (**System Administration > Configuration > SQL and Object Settings > User DDL Mappings**), you can map SQL datatypes to their InterSystems IRIS equivalents. See the [User Datatypes](#) reference for more information about this section. For more information about SQL datatypes, see the [Data Types](#) chapter in *InterSystems SQL Reference*.

## A.2 National Language Settings

The **National Language Settings** menu (**System Administration > Configuration > National Language Settings**) contains pages where you can configure national language locale definitions. For more information, see the [Configuring National Language Support](#) section of the “Configuring InterSystems IRIS” chapter in *System Administration Guide*.

## A.3 Device Settings

The **Device Settings** menu (**System Administration > Configuration > Device Settings**) contains pages where you can configure devices.

### A.3.1 Devices

Define device types. See the [Devices](#) reference page.

### A.3.2 Device Subtypes

Define device subtypes. See the [DeviceSubTypes](#) reference page.

### A.3.3 IO Settings

On the **IO Settings** page (**System Administration > Configuration > Device Settings > IO Settings**), you can set the mnemonic for WRITE commands to the following locations:

Write location	Equivalent CPF parameter
File	<a href="#">File</a>
Other	<a href="#">Other</a>
Terminal	<a href="#">Terminal</a>

The default mnemonic for each location is ^%X364.

### A.3.4 Telnet Settings

InterSystems IRIS Telnet settings apply only to Windows configurations in which InterSystems supplies the Telnet servers. They do not apply to UNIX® or 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:

Telnet Setting	Description	Equivalent CPF parameter
DNS Lookup	Allow DNS lookup of the Telnet client address.	<a href="#">DNSLookup</a>
Telnet Port Number	Set the Telnet port number.	<a href="#">Port</a>

# A.4 Additional Settings

The **Additional Settings** menu (**System Administration > Configuration > Additional Settings**) contains the following pages:

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

## A.4.1 Compatibility Settings

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

### **AsyncDisconnectErr**

Allow processes to receive disconnect errors asynchronously. This setting corresponds to the [AsyncDisconnectErr](#) parameter in the CPF.

### **AsynchError**

Allow processes to receive asynchronous errors. This setting corresponds to the [AsynchError](#) parameter in the CPF.

### **BreakMode**

Specify programmer mode response to the BREAK command. This setting corresponds to the [BreakMode](#) parameter in the CPF.

### **CollectResourceStats**

Allow InterSystems IRIS to collect system resource statistics. This setting corresponds to the [CollectResourceStats](#) parameter in the CPF.

### **DisconnectErr**

Specify how processes respond to a disconnect. This setting corresponds to the [DisconnectErr](#) parameter in the CPF.

### **FileMode**

Allow writing to a non-existent file. This setting corresponds to the [FileMode](#) parameter in the CPF.

### **GlobalKillEnabled**

Allow KILL of an unsubscribed global. This setting corresponds to the [GlobalKillEnabled](#) parameter in the CPF.

### **IEEEError**

Specify whether \$DOUBLE returns INF and NAN values system-wide. This setting corresponds to the [IEEEError](#) parameter in the CPF.

**LineRecall**

Allow command line recall for READ commands. This setting corresponds to the [LineRecall](#) parameter in the CPF.

**LogRollback**

Allow logging for transaction rollbacks. This setting corresponds to the [LogRollback](#) parameter in the CPF.

**MVDefined**

Not in use.

**NodeNameInPid**

Specify behavior when InterSystems IRIS references the special variable \$JOB. This setting corresponds to the [NodeNameInPid](#) parameter in the CPF.

**NullSubscripts**

Allow null subscripts on global references. This setting corresponds to the [NullSubscripts](#) parameter in the CPF.

**OldZU5**

Specify whether to clear global vectors when switching namespace. This setting corresponds to the [OldZU5](#) parameter in the CPF.

**OpenMode**

Specify read/write mode to use when opening sequential files. This setting corresponds to the [OpenMode](#) parameter in the CPF.

**PopError**

Specify when to pop error handlers off the stack. This setting corresponds to the [PopError](#) parameter in the CPF.

**RefInKind**

Specify how \$NAME and \$QUERY handle extended global references. This setting corresponds to the [RefInKind](#) parameter in the CPF.

**ScientificNotation**

Allow lowercase "e" as scientific notation symbol system-wide. This setting corresponds to the [ScientificNotation](#) parameter in the CPF.

**SetZEOF**

Specify the behavior when reading a sequential file and encountering an unexpected end-of-file error. This setting corresponds to the [SetZEOF](#) parameter in the CPF.

**ShutDownLogErrors**

Allow writing of InterSystems IRIS system error log entries to the messages.log file on shutdown. This setting corresponds to the [ShutDownLogErrors](#) parameter in the CPF.

**StopID**

Not in use.

**SwitchOSdir**

Disallow switching current working directories when changing namespaces. This setting corresponds to the [SwitchOSdir](#) parameter in the CPF.

**SynchCommit**

Disable synchronizing TCOMMIT with the corresponding journal write operation. This setting corresponds to the [SynchCommit](#) parameter in the CPF.

**TelnetNUL**

Suppress Telnet NUL at end-of-line for Telnet transmission. Windows systems only. This setting corresponds to the [TelnetNUL](#) parameter in the CPF.

**TruncateOverflow**

Suppress the <MAXNUMBER> error on numeric overflow. This setting corresponds to the [TruncateOverflow](#) parameter in the CPF.

**Undefined**

Specify the response when ObjectScript attempts to fetch a variable that does not exist. This setting corresponds to the [Undefined](#) parameter in the CPF.

**UseNagleAlgorithm**

Allow InterSystems IRIS to use the Nagle algorithm for Telnet. This setting corresponds to the [UseNagleAlgorithm](#) parameter in the CPF.

**ViewPastData**

Allow \$VIEW to examine data outside of InterSystems IRIS memory area. This setting corresponds to the [View-PastData](#) parameter in the CPF.

**ZDateNull**

Specify the \$ZDATE response to an invalid value. This setting corresponds to the [ZDateNull](#) parameter in the CPF.

## A.4.2 Advanced Memory

The **Advanced Memory** page ([System Administration > Configuration > Additional Settings > Advanced Memory](#)) contains the following memory-related setting:

**Important:** To allocate memory to routine and database caches, visit the [Memory and Startup](#) page ([System Administration > Configuration > System Configuration > Memory and Startup](#)). When first installed, InterSystems IRIS® data platform sets this cache allocation to happen **Automatically**. *This automatic 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 InterSystems IRIS” chapter in *System Administration Guide*.

**BackoffDisabled**

This setting corresponds to the [BackoffDisabled \(memlock\)](#) flag in the CPF.

**ConsoleFile**

Set the path to the messages log file (messages.log). This setting corresponds to the [console](#) parameter in the CPF.

**LargePagesDisabled**

This setting corresponds to the [LargePagesDisabled \(memlock\)](#) flag in the CPF.

**LargePagesRequired**

This setting corresponds to the [LargePagesRequired \(memlock\)](#) flag in the CPF.

**LibPath**

Set the LD\_LIBRARY\_PATH environment variable. UNIX® systems only. This setting corresponds to the [LibPath](#) parameter in the CPF.

**LineRecallBuffer**

This setting corresponds to the [LineRecallBuffer \(history\)](#) option in the CPF.

**LineRecallEntries**

This setting corresponds to the [LineRecallEntries \(history\)](#) option in the CPF.

**LockSharedMemory**

This setting corresponds to the [LockSharedMemory \(memlock\)](#) flag in the CPF.

**LockTextSegment**

This setting corresponds to the [LockTextSegment \(memlock\)](#) flag in the CPF.

**MaxServerConn**

Set the maximum number of connections from ECP clients. This setting corresponds to the [MaxServerConn](#) parameter in the CPF.

**MaxServers**

Set the maximum number of connections to ECP servers. This setting corresponds to the [MaxServers](#) parameter in the CPF.

**Path**

Add directories to the default path environment variable. UNIX® systems only. This setting corresponds to the [Path](#) parameter in the CPF.

**VMSConsoleTerminal**

Not in use.

**ZFSize**

This setting corresponds to the [ZFSize \(zfheap\)](#) option in the CPF.

**ZFString**

This setting corresponds to the [ZFString \(zfheap\)](#) option in the CPF.

**erlog**

Set the maximum number of entries in the error log. This setting corresponds to the [erlog](#) parameter in the CPF.

**gmheap**

Set the size of the generic memory heap. This setting corresponds to the [gmheap](#) parameter in the CPF.

**ijcbuf**

Set the size of an InterJob Communication buffer. This setting corresponds to the [ijcbuf](#) parameter in the CPF.

**ijcnum**

Set the number of InterJob Communication devices. This setting corresponds to the [ijcnum](#) parameter in the CPF.

**jrnbufs**

Allocate memory for journal buffers. This setting corresponds to the [jrnbufs](#) parameter in the CPF.

**locksiz**

Set the maximum size of shared memory for locks. This setting corresponds to the [locksiz](#) parameter in the CPF.

**netjob**

Allow remote job requests. This setting corresponds to the [netjob](#) parameter in the CPF.

**nlstab**

Set the number of NLS collation tables. This setting corresponds to the [nlstab](#) parameter in the CPF.

**targwijsz**

Set the desired size of the wij. This setting corresponds to the [targwijsz](#) parameter in the CPF.

**udevtabsiz**

Set the maximum size of the device table. This setting corresponds to the [udevtabsiz](#) parameter in the CPF.

## A.4.3 Monitor Settings

Allow automatic Simple Network Management Protocol (SNMP) startup. For more information about this option, see the [SNMPEnabled](#) reference page. For more information about SNMP, see [Monitoring InterSystems IRIS Using SNMP](#) in the *Monitoring Guide*.

## A.4.4 Source Control

For information on source control, see [Integrating InterSystems IRIS with Source Control Systems](#).

## A.4.5 Startup Settings

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

**CallinHalt**

Allow custom routines during callin close. This setting corresponds to the [CallinHalt](#) parameter in the CPF.

**CallinStart**

Allow custom routines during callin initialization. This setting corresponds to the [CallinStart](#) parameter in the CPF.

**CliSysName**

Set the node name for the local system. This setting corresponds to the [CliSysName](#) parameter in the CPF.

**DBSizesAllowed**

Specify possible database block sizes when creating a database. This setting corresponds to the [DBSizesAllowed](#) parameter in the CPF.

**DefaultPortBindAddress**

Set the port number for the InterSystems IRIS superserver. This setting corresponds to the [DefaultPortBindAddress](#) parameter in the CPF.

**EnableVSSBackup**

Allow Volume Shadow Copy Service (VSS) backup. Windows systems only. This setting corresponds to the [EnableVSSBackup](#) parameter in the CPF.

**EnsembleAutoStart**

Allow productions to auto-start when InterSystems IRIS starts. This setting corresponds to the [EnsembleAutoStart](#) parameter in the CPF.

**ErrorPurge**

Set the number of days to keep error globals. This setting corresponds to the [ErrorPurge](#) parameter in the CPF.

**FIPSMode**

Enable FIPS 140–2 compliant library for database encryption on Red Hat Linux. This setting corresponds to the [FIPSMode](#) parameter in the CPF.

**IPv6**

Allow InterSystems IRIS to accept IPv6 addresses. This setting corresponds to the [IPv6](#) parameter in the CPF.

**JobHalt**

Allow custom routines during background process (job) shutdown. This setting corresponds to the [JobHalt](#) parameter in the CPF.

**JobServers**

Set the number of job servers. This setting corresponds to the [JobServers](#) parameter in the CPF.

**JobStart**

Allow custom routines during background process (job) startup. This setting corresponds to the [JobStart](#) parameter in the CPF.

**LicenseID**

Allow instance to requests a key from the license server. This setting corresponds to the [LicenseID](#) parameter in the CPF.

**MaxConsoleLogSize**

Set the maximum size of the messages.log file. This setting corresponds to the [MaxConsoleLogSize](#) parameter in the CPF.

**MaxIRISTempSizeAtStart**

Set the maximum size of the IRISTEMP database on restart. This setting corresponds to the [MaxIRISTempSizeAtStart](#) parameter in the CPF.

**ProcessHalt**

Allow custom routines during foreground process shutdown. This setting corresponds to the [ProcessHalt](#) parameter in the CPF.

**ProcessStart**

Allow custom routines during foreground process startup. This setting corresponds to the [ProcessStart](#) parameter in the CPF.

**ShutdownTimeout**

Set the number of seconds InterSystems IRIS should wait until forcing a shutdown. This setting corresponds to the [ShutdownTimeout](#) parameter in the CPF.

**SystemHalt**

Allow custom routines during system shutdown. This setting corresponds to the [SystemHalt](#) parameter in the CPF.

**SystemStart**

Allow custom routines during system startup. This setting corresponds to the [SystemStart](#) parameter in the CPF.

**TempDirectory**

Specify the subdirectory for temporary files. This setting corresponds to the [TempDirectory](#) parameter in the CPF.

**TerminalPrompt**

Define the format of the terminal prompt. This setting corresponds to the [TerminalPrompt](#) parameter in the CPF.

**WebServer**

Allow the web server to start. This setting corresponds to the [WebServer](#) parameter in the CPF.

**WebServerName**

Set an IP address or a DNS name for the Web server. This setting corresponds to the [WebServerName](#) parameter in the CPF.

**WebServerPort**

Set the web server port. This setting corresponds to the [WebServerPort](#) parameter in the CPF.

**WebServerURLPrefix**

Identify the InterSystems IRIS instance name for the web server to access. This setting corresponds to the [WebServerURLPrefix](#) parameter in the CPF.

## ZSTU

Allow the user-defined startup to run. This setting corresponds to the **ZSTU** parameter in the CPF.

## A.4.6 Task Manager Email

See the [Configuring Task Manager Email Settings](#) section of the “Configuring InterSystems IRIS” chapter in *System Administration Guide*.