



# Configuration Parameter File Reference

Version 2024.2  
2024-09-05

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, and TrakCare are registered trademarks of InterSystems Corporation. HealthShare® CMS Solution Pack™, HealthShare® Health Connect Cloud™, InterSystems® Data Fabric Studio™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, and InterSystems TotalView™ For Asset Management are trademarks of InterSystems Corporation. TrakCare is a registered trademark in Australia and the European Union.

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

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

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

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

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

For Support questions about any InterSystems products, contact:

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

# Table of Contents

<b>1 Introduction to the Configuration Parameter File .....</b>	<b>1</b>
1.1 CPF Overview .....	1
1.1.1 CPF Format .....	1
1.1.2 Sample CPF .....	2
1.2 Editing the Active CPF .....	7
1.2.1 Choosing a CPF at Startup .....	7
1.3 Using the Configuration Merge Feature .....	8
1.3.1 Configuration Merge Example .....	8
1.4 Configuration Security .....	9
1.5 Parameter Descriptions .....	10
<b>[Actions] .....</b>	<b>11</b>
CreateApplication .....	12
DeleteApplication .....	13
ModifyApplication .....	14
Execute .....	15
CreateComport .....	16
DeleteComport .....	17
ModifyComport .....	18
ModifyConfig .....	19
CreateDatabase .....	20
DeleteDatabase .....	21
ModifyDatabase .....	22
CreateDatabaseFile .....	23
DeleteDatabaseFile .....	24
ModifyDatabaseFile .....	25
CreateDevice .....	26
DeleteDevice .....	27
ModifyDevice .....	28
CreateDeviceSubType .....	29
DeleteDeviceSubType .....	30
ModifyDeviceSubType .....	31
CreateDocDB .....	32
DeleteDocDB .....	33
ModifyDocDB .....	34
ModifyECP .....	35
CreateECPServer .....	36
DeleteECPServer .....	37
ModifyECPServer .....	38
CreateEvent .....	39
DeleteEvent .....	40
ModifyEvent .....	41
ModifyJournal .....	42
CreateLDAPConfig .....	43
DeleteLDAPConfig .....	44
ModifyLDAPConfig .....	45
CreateLicenseServer .....	46
DeleteLicenseServer .....	47

ModifyLicenseServer .....	48
CreateMagTapes .....	49
DeleteMagTapes .....	50
ModifyMagTapes .....	51
CreateMapGlobal .....	52
DeleteMapGlobal .....	53
ModifyMapGlobal .....	54
CreateMapPackage .....	55
DeleteMapPackage .....	56
ModifyMapPackage .....	57
CreateMapRoutine .....	58
DeleteMapRoutine .....	59
ModifyMapRoutine .....	60
ConfigMirror .....	61
CreateMirror .....	62
DeleteMirror .....	63
ModifyMirror .....	64
ModifyMirrorMember .....	65
ModifyMiscellaneous .....	66
ModifyMonitor .....	67
CreateNamespace .....	68
DeleteNamespace .....	69
ModifyNamespace .....	70
CreateResource .....	71
DeleteResource .....	72
ModifyResource .....	73
CreateRole .....	74
DeleteRole .....	75
ModifyRole .....	76
ModifyService .....	77
ConfigShardedCluster .....	78
GrantPrivilege .....	79
RevokePrivilege .....	80
ModifySQL .....	81
CreateSqlSysDatatype .....	82
DeleteSqlSysDatatype .....	83
ModifySqlSysDatatype .....	84
CreateSqlUserDatatype .....	85
DeleteSqlUserDatatype .....	86
ModifySqlUserDatatype .....	87
CreateSSLConfig .....	88
DeleteSSLConfig .....	89
ModifySSLConfig .....	90
ModifyStartup .....	91
CreateServer .....	92
DeleteServer .....	93
ModifyServer .....	94
ModifySystem .....	95
CreateUser .....	96
DeleteUser .....	97
ModifyUser .....	98

CreateWorkQueue .....	99
DeleteWorkQueue .....	100
ModifyWorkQueue .....	101
<b>[Archives] .....</b>	<b>103</b>
Archives .....	104
<b>[ComPorts] .....</b>	<b>107</b>
COMn .....	108
<b>[config] .....</b>	<b>111</b>
LibPath .....	112
MaxServerConn .....	113
MaxServers .....	114
Path .....	115
PythonPath .....	116
PythonRuntimeLibrary .....	117
PythonRuntimeLibraryVersion .....	118
UUIDv1RandomMac .....	119
bbsiz .....	120
console .....	121
errlog .....	122
globals .....	123
gmheap .....	124
history .....	125
ijcbuff .....	126
ijcnum .....	127
jrnbufs .....	128
locksiz .....	129
memlock .....	130
netjob .....	132
nlstab .....	133
overview .....	134
pijdir .....	135
routines .....	136
udevtabsiz .....	138
wijdir .....	139
targwijsz .....	140
zfheap .....	141
<b>[ConfigFile] .....</b>	<b>143</b>
Version .....	144
<b>[Databases] .....</b>	<b>145</b>
Database .....	146
<b>[Debug] .....</b>	<b>149</b>
Dumpstyle .....	150
Semsperset .....	152
<b>[Devices] .....</b>	<b>153</b>
Devices .....	154
<b>[DeviceSubTypes] .....</b>	<b>157</b>
DeviceSubTypes .....	158

<b>[ECP]</b> .....	<b>161</b>
ClientReconnectDuration .....	162
ClientReconnectInterval .....	163
ServerTroubleDuration .....	164
<b>[ECPServers]</b> .....	<b>165</b>
ECPServers .....	166
<b>[Gateways]</b> .....	<b>167</b>
%DotNet Server .....	168
%IntegratedML Server .....	170
%JDBC Server .....	171
%Java Server .....	172
%Python Server .....	173
%R Server .....	174
%XSLT Server .....	175
Remote Server .....	176
<b>[IO]</b> .....	<b>177</b>
Other .....	178
File .....	179
Terminal .....	180
<b>[Journal]</b> .....	<b>181</b>
AlternateDirectory .....	182
ArchiveName .....	183
BackupsBeforePurge .....	184
CompressFiles .....	185
CurrentDirectory .....	186
DaysBeforePurge .....	187
FileSizeLimit .....	188
FreezeOnError .....	189
JournalFilePrefix .....	190
JournalcspSession .....	191
PurgeArchive .....	192
<b>[LicenseServers]</b> .....	<b>193</b>
LicenseServers .....	194
<b>[Logging]</b> .....	<b>195</b>
ChildProcessLaunchCommand .....	196
Enabled .....	197
Format .....	198
Interval .....	199
Level .....	200
<b>[Map]</b> .....	<b>201</b>
Global .....	202
Package .....	204
Routine .....	205
<b>[MapMirrors]</b> .....	<b>207</b>
MapMirrors .....	208
<b>[MirrorMember]</b> .....	<b>211</b>
AgentAddress .....	212

AsyncMemberGUID .....	213
AsyncMemberType .....	214
AsyncUseSystemPurgeInterval .....	215
JoinMirror .....	216
SystemName .....	217
ValidatedMember .....	218
VirtualAddressInterface .....	219
<b>[Mirrors] .....</b>	<b>221</b>
Mirrors .....	222
<b>[Miscellaneous] .....</b>	<b>225</b>
AsyncDisconnectErr .....	226
AsynchError .....	227
BreakMode .....	228
CollectResourceStats .....	229
DisconnectErr .....	230
FileMode .....	231
GlobalKillEnabled .....	232
IEEEError .....	233
LicenseAltHeaders .....	234
LineRecall .....	235
ListFormat .....	236
LogRollback .....	237
MVDefined .....	238
NodeNameInPid .....	239
NullSubscripts .....	240
OldZU5 .....	241
OpenMode .....	242
PopError .....	243
RefInKind .....	244
ScientificNotation .....	245
SetZEOF .....	246
ShutDownLogErrors .....	247
StopID .....	248
SwitchOSdir .....	249
SynchCommit .....	250
TelnetNUL .....	251
TruncateOverflow .....	252
Undefined .....	253
UseNagleAlgorithm .....	254
ViewPastData .....	255
ZDateNull .....	256
<b>[Monitor] .....</b>	<b>257</b>
SNMPEnabled .....	258
<b>[Namespaces] .....</b>	<b>259</b>
Namespace .....	260
<b>[SQL] .....</b>	<b>263</b>
ANSIPrecedence .....	264
AdaptiveMode .....	265

AllowRowIDUpdate .....	266
AutoParallel .....	267
AutoParallelThreshold .....	268
BiasQueriesAsOutlier .....	269
ClientMaxIdleTime .....	270
Comment .....	271
DBMSSecurity .....	272
DDLDefineBitmapExtent .....	273
DDLFinal .....	274
DDLNo201 .....	275
DDLNo30 .....	276
DDLNo307 .....	277
DDLNo311 .....	278
DDLNo315 .....	279
DDLNo324 .....	280
DDLNo333 .....	281
DDLSQLOnlyCompile .....	282
DDLUseExtentSet .....	283
DDLUseSequence .....	284
DefaultSchema .....	285
DelimitedIds .....	286
DropDelete .....	287
ECPSync .....	288
ExtrinsicFunctions .....	289
FastDistinct .....	290
IdKey .....	291
IdTrxFrom .....	292
IdTrxTo .....	293
LockThreshold .....	294
LockTimeout .....	295
ODBCVarcharMaxlen .....	296
ParameterSampling .....	297
QueryProcedures .....	298
RTPC .....	299
ReferentialChecks .....	300
SaveMAC .....	301
TCPKeepAlive .....	302
TODATEDefaultFormat .....	303
TimePrecision .....	304
<b>[SqlSysDatatypes] .....</b>	<b>305</b>
System Datatypes .....	306
<b>[SqlUserDatatypes] .....</b>	<b>307</b>
User Datatypes .....	308
<b>[Startup] .....</b>	<b>309</b>
CallinHalt .....	310
CallinStart .....	311
CliSysName .....	312
DBSizesAllowed .....	313
DefaultPort .....	314



DefaultPortBindAddress .....	315
EnableSharding .....	316
EnableVSSBackup .....	317
EnsembleAutoStart .....	318
ErrorPurge .....	319
FIPSMODE .....	320
IPv6 .....	321
JobHalt .....	322
JobServers .....	323
JobStart .....	326
LicenseID .....	327
MaxConsoleLogSize .....	328
MaxIRISTempSizeAtStart .....	329
PasswordHash .....	330
ProcessHalt .....	331
ProcessStart .....	332
ShutdownTimeout .....	333
SystemHalt .....	334
SystemMode .....	335
SystemStart .....	336
TempDirectory .....	337
TerminalPrompt .....	338
WebServer .....	339
WebServerName .....	340
WebServerPort .....	341
WebServerProtocol .....	342
WebServerSSLConfiguration .....	343
WebServerURLPrefix .....	344
ZSTU .....	345
<b>[Telnet] .....</b>	<b>347</b>
DNSLookup .....	348
Port .....	349
<b>[WorkQueues] .....</b>	<b>351</b>
Default .....	352
SQL .....	353

# List of Tables

Table X-1: SQL Tab ..... 263

# 1

## Introduction to the Configuration Parameter File

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

This topic discusses how to use and edit the CPF. The [Table of Contents](#) at the beginning of this reference shows each parameter in the CPF, sorted by section.

**Note:** Within this document and the CPF itself, memory is depicted 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 [Editing the Active CPF](#).

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 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.

**Important:** Beginning with version 2022.1.1 of InterSystems IRIS, setting a parameter equal to empty quotes (*parameter=""*) sets its value to the empty string, or no value. (In previous versions, doing so set the parameter to its default value.) You cannot set a required parameter to the empty string, so setting such a parameter to empty quotes results in an error.

### 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 comment and “\*/” (asterisk, slash) to end it.

## 1.1.2 Sample CPF

A sample default CPF, installed on a Windows system as part of InterSystems IRIS version 2024.1 with an auto-configured web server, is shown in the following. Note, for example, the default superserver and web server ports (1972 and 80, respectively) in the values for `DefaultPort` and `WebServerPort` parameters in the `Startup` section.

```
[ConfigFile]  
Product=IRIS  
Version=2024.2
```

```
[Databases]
IRISSYS=C:\InterSystems\IRIS\mgr\
IRISLIB=C:\InterSystems\IRIS\mgr\irislib\
IRISTEMP=C:\InterSystems\IRIS\mgr\iristemp\
IRISLOCALDATA=C:\InterSystems\IRIS\mgr\irislocaldata\
IRISAUDIT=C:\InterSystems\IRIS\mgr\irisaudit\
ENSLIB=C:\InterSystems\IRIS\mgr\enslib\
USER=C:\InterSystems\IRIS\mgr\user\

[Namespaces]
%SYS=IRISSYS
USER=USER

[MirrorMember]
AgentAddress=
AsyncMemberGUID=
AsyncMemberType=0
AsyncUseSystemPurgeInterval=0
JoinMirror=0
SystemName=
ValidatedMember=0
VirtualAddressInterface=

[Journal]
AlternateDirectory=C:\InterSystems\IRIS\mgr\journal\
ArchiveName=
BackupsBeforePurge=2
CompressFiles=1
CurrentDirectory=C:\InterSystems\IRIS\mgr\journal\
DaysBeforePurge=2
FileSizeLimit=1024
FreezeOnError=0
JournalFilePrefix=
JournalcspSession=0
PurgeArchived=0

[Startup]
CallinHalt=1
CallinStart=1
CliSysName=
DBSizesAllowed=8192
DefaultPort=1972
DefaultPortBindAddress=
EnableVSSBackup=1
EnsembleAutoStart=1
ErrorPurge=30
FIPSMODE=0
IPv6=0
JobHalt=1
JobServers=0
JobStart=1
LicenseID=
MaxConsoleLogSize=5
MaxIRISTempSizeAtStart=0
PasswordHash=
ProcessHalt=1
ProcessStart=1
ShutdownTimeout=300
SystemHalt=1
SystemMode=
SystemStart=1
TempDirectory=Temp
TerminalPrompt=8,2
WebServer=0
WebServerName=
WebServerPort=80
WebServerProtocol=http
WebServerSSLConfiguration=
WebServerURLPrefix=iris
ZSTU=1

[WorkQueues]
Default=
SQL=
Utility=12,12,,,1

[Logging]
ChildProcessLaunchCommand=irislogd.exe -f /tmp/irislogd.log
Enabled=0
Format=NVP
Interval=10
Level=WARN

[Gateways]
%DotNet Server=.NET,53372,%Gateway_Object,N6.0
```

```

%IntegratedML Server=ML,53572,%Gateway_ML
%JDBC Server=JDBC,53772,%Gateway_SQL,,,,,0
%Java Server=Java,53272,%Gateway_Object
%Python Server=Python,53472,%Gateway_Object
%R Server=R,53872,%Gateway_Object
%XSLT Server=XSLT,53672,%Gateway_Object,,,,,0

[DeviceSubTypes]
C-ANSI=80^#,$C(27,91,72,27,91,74)^25^$C(8)^W $C(27,91)_(DY+1)_"_";_"_(DX+1)_"H" S
$X=DX,$Y=DY^$C(27,91,74)^$C(27,91,75)^$C(27,91,72,27,91,74)^$C(8,32,8)
C-IRIS Terminal=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-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)^^^^

[Devices]
0=0^TRM^C-IRIS Terminal^^^^Principal device^
2=2^SPL^PK-DEC^^^^Spool LA120^
47=47^MT^M/UX^^("auv":0:2048)^^Magnetic tape^
48=48^MT^M/UX^^("avl":0:2048)^^Magnetic tape^
57=57^BT^M/UX^^("auv":0:2048)^^Magnetic tape^
58=58^BT^M/UX^^("avl":0:2048)^^Magnetic tape^
SPOOL=2^SPL^PK-DEC^^^^Spool LA120^
TERM=0^TRM^C-IRIS Terminal^^^^Windows Console^
|LAT|=0^TRM^C-VT220^^^^Principal device^
|PRN|=|PRN|^OTH^P-DEC^^"W"^^Windows Printer^
|TNT|=0^TRM^C-VT220^^^^Principal device^
|TRM|=0^TRM^C-IRIS Terminal^^^^Windows Console^

[MagTapes]
47=\\.\\TAPE0
48=\\.\\TAPE1
57=\\.\\TAPE0
58=\\.\\TAPE1

[config]
LibPath=
MaxServerConn=1
MaxServers=2
Path=
PythonPath=
PythonRuntimeLibrary=
PythonRuntimeLibraryVersion=
UIDv1RandomMac=0
bbsiz=-1
console=,
errlog=500
globals=0,0,0,0,0,0
gmheap=0
history=500
ijcbuf=512
ijcnum=16
jrnbufs=64
locksiz=0
memlock=0
netjob=1
nlstab=50
overview=Windows (Intel)~Windows
pijdir=
routines=0
targwijsz=0
udevtabsiz=24576
wijdir=
zfheap=0,0

[Miscellaneous]
AsyncDisconnectErr=0
AsynchError=1
BreakMode=1
CollectResourceStats=0

```

```

DisconnectErr=0
FileMode=0
GlobalKillEnabled=1
IEEEError=1
LicenseAltHeaders=0
LineRecall=1
ListFormat=0
LogRollback=0
MVDefined=0
NodeNameInPid=0
NullSubscripts=0
OldZU5=0
OpenMode=0
PopError=0
RefInKind=0
ScientificNotation=1
SetZEOF=0
ShutDownLogErrors=0
StopID=0
SwitchOSdir=0
SynchCommit=0
TelnetNUL=0
TruncateOverflow=0
Undefined=0
UseNagleAlgorithm=0
ViewPastData=0
ZDateNull=0
ZaMode=0

[ECP]
ClientReconnectDuration=1200
ClientReconnectInterval=5
ServerTroubleDuration=60

[Cluster]
CommIPAddress=
JoinCluster=0

[LicenseServers]
LOCAL=127.0.0.1,4002

[Monitor]
SNMPEnabled=0

[IO]
File=%X364
MagTape=%XMAG
Other=%X364
Terminal=%X364

[SQL]
ANSIPrecedence=1
AdaptiveMode=1
AllowRowIDUpdate=0
AutoParallel=1
AutoParallelThreshold=3200
BiasQueriesAsOutlier=0
ClientMaxIdleTime=0
Comment=1
DBMSSecurity=1
DDLDefineBitmapExtent=1
DDLFinal=1
DDLNo201=0
DDLNo30=0
DDLNo307=0
DDLNo311=0
DDLNo315=0
DDLNo324=0
DDLNo333=0
DDLSQLOnlyCompile=0
DDLUseExtentSet=1
DDLUseSequence=1
DefaultSchema=SQLUser
DelimitedIds=1
DropDelete=1
ECPSync=0
ExtrinsicFunctions=0
FastDistinct=1
IdKey=1
IdTrxFrom=~ `!@#$$%^&*( )_+--[ ]\{|;':",./<>?
IdTrxTo=
LockThreshold=1000
LockTimeout=10
ODBCVarcharMaxlen=4096
ParameterSampling=0

```

```

QueryProcedures=0
RTPC=1
ReferentialChecks=1
SaveMAC=0
TCPKeepAlive=300
TODATEDefaultFormat=DD MON YYYY
TimePrecision=0

[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
BLOB=%Stream.GlobalBinary
CHAR=%Library.String(MAXLEN=1)
CHAR VARYING=%Library.String(MAXLEN=1)
CHAR VARYING(%1)=%Library.String(MAXLEN=%1)
CHAR(%1)=%Library.String(MAXLEN=%1)
CHARACTER=%Library.String(MAXLEN=1)
CHARACTER VARYING=%Library.String(MAXLEN=1)
CHARACTER VARYING(%1)=%Library.String(MAXLEN=%1)
CHARACTER(%1)=%Library.String(MAXLEN=%1)
CLOB=%Stream.GlobalCharacter
DATE=%Library.Date
DATETIME=%Library.DateTime
DATETIME2=%Library.DateTime
DEC=%Library.Numeric(MAXVAL=9999999999999999,MINVAL=-9999999999999999,SCALE=0)
DEC(%1)=%Library.Numeric(MAXVAL=<| '$$maxval^%apiSQL(%1,0)' |>,MINVAL=<| '$$minval^%apiSQL(%1,0)' |>,SCALE=0)
DEC(%1,%2)=%Library.Numeric(MAXVAL=<| '$$maxval^%apiSQL(%1,%2)' |>,MINVAL=<| '$$minval^%apiSQL(%1,%2)' |>,SCALE=%2)
DECIMAL=%Library.Numeric(MAXVAL=9999999999999999,MINVAL=-9999999999999999,SCALE=0)
DECIMAL(%1)=%Library.Numeric(MAXVAL=<| '$$maxval^%apiSQL(%1,0)' |>,MINVAL=<| '$$minval^%apiSQL(%1,0)' |>,SCALE=0)
DECIMAL(%1,%2)=%Library.Numeric(MAXVAL=<| '$$maxval^%apiSQL(%1,%2)' |>,MINVAL=<| '$$minval^%apiSQL(%1,%2)' |>,SCALE=%2)
DOUBLE=%Library.Double
DOUBLE PRECISION=%Library.Double
EMBEDDING(%1,%2)=%Library.Embedding(MODEL=%1,SOURCE=%2)
FLOAT=%Library.Double
FLOAT(%1)=%Library.Double
IMAGE=%Stream.GlobalBinary
INT=%Library.Integer(MAXVAL=2147483647,MINVAL=-2147483648)
INT(%1)=%Library.Integer(MAXVAL=2147483647,MINVAL=-2147483648)
INTEGER=%Library.Integer(MAXVAL=2147483647,MINVAL=-2147483648)
LONG=%Stream.GlobalCharacter
LONG BINARY=%Stream.GlobalBinary
LONG RAW=%Stream.GlobalBinary
LONG VARCHAR=%Stream.GlobalCharacter
LONG VARCHAR(%1)=%Stream.GlobalCharacter
LONGTEXT=%Stream.GlobalCharacter
LONGVARBINARY=%Stream.GlobalBinary
LONGVARBINARY(%1)=%Stream.GlobalBinary
LONGVARCHAR=%Stream.GlobalCharacter
LONGVARCHAR(%1)=%Stream.GlobalCharacter
MEDIUMINT=%Library.Integer(MAXVAL=8388607,MINVAL=-8388608)
MEDIUMINT(%1)=%Library.Integer(MAXVAL=8388607,MINVAL=-8388608)
MEDIUMTEXT=%Stream.GlobalCharacter
MONEY=%Library.Currency
NATIONAL CHAR=%Library.String(MAXLEN=1)
NATIONAL CHAR VARYING=%Library.String(MAXLEN=1)
NATIONAL CHAR VARYING(%1)=%Library.String(MAXLEN=%1)
NATIONAL CHAR(%1)=%Library.String(MAXLEN=%1)
NATIONAL CHARACTER=%Library.String(MAXLEN=1)
NATIONAL CHARACTER VARYING=%Library.String(MAXLEN=1)
NATIONAL CHARACTER VARYING(%1)=%Library.String(MAXLEN=%1)
NATIONAL CHARACTER(%1)=%Library.String(MAXLEN=%1)
NATIONAL VARCHAR=%Library.String(MAXLEN=1)
NATIONAL VARCHAR(%1)=%Library.String(MAXLEN=%1)
NCHAR=%Library.String(MAXLEN=1)
NCHAR(%1)=%Library.String(MAXLEN=%1)
NTEXT=%Stream.GlobalCharacter
NUMBER=%Library.Numeric(SCALE=0)
NUMBER(%1)=%Library.Numeric(MAXVAL=<| '$$maxval^%apiSQL(%1)' |>,MINVAL=<| '$$minval^%apiSQL(%1)' |>,SCALE=0)
NUMBER(%1,%2)=%Library.Numeric(MAXVAL=<| '$$maxval^%apiSQL(%1,%2)' |>,MINVAL=<| '$$minval^%apiSQL(%1,%2)' |>,SCALE=%2)
NUMERIC=%Library.Numeric(MAXVAL=9999999999999999,MINVAL=-9999999999999999,SCALE=0)
NUMERIC(%1)=%Library.Numeric(MAXVAL=<| '$$maxval^%apiSQL(%1,0)' |>,MINVAL=<| '$$minval^%apiSQL(%1,0)' |>,SCALE=0)
NUMERIC(%1,%2)=%Library.Numeric(MAXVAL=<| '$$maxval^%apiSQL(%1,%2)' |>,MINVAL=<| '$$minval^%apiSQL(%1,%2)' |>,SCALE=%2)
NVARCHAR=%Library.String(MAXLEN=1)
NVARCHAR(%1)=%Library.String(MAXLEN=%1)
NVARCHAR(%1,%2)=%Library.String(MAXLEN=%1)
NVARCHAR(MAX)=%Stream.GlobalCharacter
POSIXTIME=%Library.PosixTime
RAW(%1)=%Library.Binary(MAXLEN=%1)
REAL=%Library.Double
ROWVERSION=%Library.RowVersion

```



```

SERIAL=%Library.Counter
SMALLDATETIME=%Library.DateTime(MINVAL="1900-01-01 00:00:00",MAXVAL="2079-06-06 23:59:59")
SMALLINT=%Library.SmallInt
SMALLINT(%1)=%Library.SmallInt
SMALLMONEY=%Library.Currency
SYSNAME=%Library.String(MAXLEN=128)
TEXT=%Stream.GlobalCharacter
TIME=%Library.Time
TIME(%1)=%Library.Time(PRECISION=%1)
TIMESTAMP=%Library.PosixTime
TIMESTAMP2=%Library.TimeStamp
TINYINT=%Library.TinyInt
TINYINT(%1)=%Library.TinyInt
UNIQUEIDENTIFIER=%Library.UniqueIdentifier
VARBINARY=%Library.Binary(MAXLEN=1)
VARBINARY(%1)=%Library.Binary(MAXLEN=%1)
VARCHAR=%Library.String(MAXLEN=1)
VARCHAR(%1)=%Library.String(MAXLEN=%1)
VARCHAR(%1,%2)=%Library.String(MAXLEN=%1)
VARCHAR(MAX)=%Stream.GlobalCharacter
VARCHAR2(%1)=%Library.String(MAXLEN=%1)
VECTOR=%Library.Vector(DATATYPE="DOUBLE")
VECTOR(%1)=%Library.Vector(DATATYPE=%1)
VECTOR(%1,%2)=%Library.Vector(DATATYPE=%1,LEN=%2)

[Telnet]
DNSLookup=ON
Port=23

[Conversions]
LastConvertTime=2024-07-16 10:15:29

```

## 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. Open the `iris.cpf` file, located in the [installation directory](#), and make the desired changes. InterSystems recommends that you save a backup copy of the CPF before editing it, as an invalid CPF may cause InterSystems IRIS to fail to start. Be sure to follow the syntax described in [CPF Format](#).

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:

- [Choosing a CPF at Startup](#)
- [Using the Configuration Merge Feature](#)

### 1.2.1 Choosing a CPF at 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:\IRIS`, you might have the following CPFs:

```

C:\IRIS\iris.cpf           ; default CPF
C:\IRIS\production.cpf    ; for production
C:\IRIS\development.cpf   ; for development
C:\IRIS\testapps.cpf      ; for testing
C:\IRIS\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](#).

At shutdown, the instance 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.

## 1.3 Using the Configuration Merge Feature

You can modify the default `iris.cpf` using a declarative *configuration merge file*. A merge file is a partial CPF that sets the desired values for any number of CPF parameters.

Configuration merge is useful for a number of purposes. A merge file lets you specify individual settings for instances deployed from the same source, supporting automated deployment and a DevOps approach. You can use configuration merge with containerized and noncontainerized InterSystems IRIS instances; for more information, see [Configuration merge in deployment](#) and [Reconfigure an existing instance using configuration merge](#) in *Automating Configuration of InterSystems IRIS with Configuration Merge*.

Configuration merge is very useful in automated deployment (on UNIX® and Linux systems only) because it makes the specified configuration changes before the new instance first starts, allowing you to customize the configurations of multiple instances deployed from the same container image or install kit. Automated deployment of multinode topologies can use multiple merge files to customize different groups of instances; for example, in automated deployment of a [sharded cluster with compute nodes](#), you would apply different merge files for data node 1, the remaining data nodes, and the compute nodes in that order, and when deploying a [mirror](#), you would apply different merge files for the primary, backup, and async members.

Automated reconfiguration of multiple instances can be achieved in the same way, by restarting groups of instances specifying the applicable merge file for each.

The **iris merge** command (available on Windows as well as UNIX and Linux systems) lets you use configuration merge to reconfigure a running instance. By automating application of the same merge file to multiple running instances using **iris merge**, you can simultaneously reconfigure all of those instances in the same way, applying the same set of configuration changes across your application or cluster. A single automated program can of course apply different merge files to different groups of instances (such as different mirror member or cluster nodes types) as described in the foregoing.

A merge file has the same syntax as a CPF (described in the [CPF Format](#) section), and can have any name and extension. The merge file may contain any of the parameters found within the CPF, though it is not necessary to include the values you are not changing.

Unlike the CPF, a merge file can contain duplicates of the same section and parameter. In this case, InterSystems IRIS prioritizes the value closer to the end of the file, which allows you to create a template merge file. For example, if you keep generally desired values at the top of the file and append instance-specific values at the bottom, InterSystems IRIS will prioritize the instance-specific values when reading the merge file.

For an example of using a merge file, see the section that follows. For more information about using a merge file when deploying InterSystems IRIS containers, see [Automating Configuration of InterSystems IRIS with Configuration Merge](#). For information about using a merge file with the InterSystems Kubernetes Operator (IKO), see [configSource: Create configuration files and a config map for them](#) in *Using the InterSystems Kubernetes Operator*.

### 1.3.1 Configuration Merge Example

This example describes how to use a merge file at startup to modify the [shared memory heap](#) and the [database cache](#) in a noncontainerized instance. These settings are controlled by the `gmheap` and `globals` parameters respectively.

The first step is to create the merge file. The example file below is named `config_merge.cpf`, though any name or extension is valid. Note that the merge file uses the same [syntax](#) as a CPF.

```
# Example configuration merge file.

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

Next, use the `iris stop` command to shut down the target instance for the merge.

```
$ sudo iris stop IRIS
```

Finally, restart the instance with `ISC_CPF_MERGE_FILE` set, as in the following script.

When this script runs, InterSystems IRIS modifies the `iris.cpf` file as specified in the `config_merge.cpf` file.

```
#!/bin/bash

# Start InterSystems IRIS with the necessary parameters (all on one line).
sudo ISC_CPF_MERGE_FILE=/merge_files/config_merge.cpf iris start IRIS
```

When the instance starts up, the merge is complete! Check that the `iris.cpf` file contains the desired values for the `gmheap` and `globals` settings.

```
[config]
...
errlog=500
globals=0,0,800,0,0,0
gmheap=256000
history=500,1024
...
```

**Note:** If the merge file specified by `ISC_CPF_MERGE_FILE` is not present, instance startup displays an error message and continues.

## 1.4 Configuration Security

To protect against accidental or intentional misconfiguration of the CPF, you can enable **Configuration Security**. This option is available on the **System-wide Security Parameters** page of the Management Portal (**System Administration > Security > System Security > System-wide Security Parameters**).

When **Configuration Security** is enabled, if InterSystems IRIS startup detects that the configuration parameter file has been modified since the last time InterSystems IRIS was started, InterSystems IRIS startup requests a username and password to validate the changes. The user account 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.

**Note:** This **Configuration Security** setting is not a substitute for operating-system–level security. InterSystems recommends that you protect the configuration file by strictly limiting the ability of users to modify it, at the operating-system level.

For more information on other system-wide security parameters, see [System Management and Security](#).

## 1.5 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.

# [Actions]

In addition to changing the values of configuration parameters, the [configuration merge feature](#) can create, modify, and delete dozens of different InterSystems IRIS objects, such as namespaces and databases, users, roles, and resource, and mirrors and mirror members, on both newly deployed and existing instances. This is done using the parameters in the [Actions] section, which is valid only in a merge file.

**Important:** Do not add the [Actions] section directly to the Configuration Parameter File (CPF). The [Actions] section is not supported in the CPF, and causes instance startup to fail if included.

The operations specified in [Actions] are idempotent, meaning that they are executed only if they would result in a change — if an object to be created exists, an object to be deleted does not exist, or an object to be modified matches the specified change, the operation is skipped. The order of the operations in the [Actions] section of a configuration merge file has no effect on the order in which they are executed; InterSystems IRIS performs the operations in a deterministic order.

For complete information on the configuration merge feature, see [Automating Configuration of InterSystems IRIS with Configuration Merge](#); for an explanation of how the [Actions] parameters are used and lists of the parameters and details about their usage, see [Can configuration merge customize more than the configuration?](#) and [\[Actions\] Parameter Reference](#), respectively, in that document.

# CreateApplication

---

Create a new application.

## Synopsis

```
[Actions]  
CreateApplication:Name=ApplicationName[,Additional Properties]
```

## Description

CreateApplication defines an application. InterSystems IRIS® creates the defined application when processing the [\[Actions\]](#) section during a configuration merge. You can specify any possible application properties in the CreateApplication definition. During the merge, InterSystems IRIS runs Security.Applications.Create() to create the application defined by CreateApplication. It only requires you to define *Name* and any additional properties needed for that type of application; the complete list of application properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateApplication by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Defining Applications](#) for more information about managing applications from the management portal.

# DeleteApplication

---

Delete an application.

## Synopsis

```
[Actions]  
DeleteApplication:ApplicationName
```

## Description

DeleteApplication deletes an application from the security database. InterSystems IRIS® deletes the defined application when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.Applications.Delete() to create the application defined by DeleteApplications. It only requires you to define *Name*.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteApplication by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Defining Applications](#) for more information about managing applications from the management portal.

# ModifyApplication

---

Modify an application.

## Synopsis

```
[Actions]  
ModifyApplication:Name=ApplicationName, [PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyApplication modifies an existing application. InterSystems IRIS® modifies the defined application when processing the [\[Actions\]](#) section during a configuration merge. You specify which application properties you want to modify in the ModifyApplication definition. During the merge, InterSystems IRIS runs Security.Applications.Modify() to modify the application defined by ModifyApplication. It only requires you to define *Name* and the properties you want to modify for that application; the complete list of application properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyApplication by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Defining Applications](#) for more information about managing applications from the management portal.



# Execute

Execute a class method or routine.

## Synopsis

```
[Actions]
Execute:Namespace=namespace,[Additional Properties]
```

## Description

`Execute` executes a class or routine. InterSystems IRIS® executes the defined class or routine when processing the [\[Actions\]](#) section during a configuration merge. The `Execute` parameters are always processed last. You specify the class or routine and its arguments in the `Execute` definition. During the merge, InterSystems IRIS runs the class or routine defined by `Execute`.

## Examples

You can execute a class or a routine with `Execute`. For example:

**Execute the class method `SYS.ClassA.MethodZ()` and pass three arguments to it; equivalent of `SYS.ClassA.MethodZ("arg1","arg2","arg3")`:**

```
Execute:Namespace="%SYS",ClassName="SYS.ClassA",MethodName="MethodZ",Arg1="arg1",Arg2="arg2",Arg3="arg3"
```

**Load and compile classes from an XML file; equivalent of `$SYSTEM.OBJ.Load("c:\iris\test.xml","ck")`:**

```
Execute:Namespace="%SYS",ClassName=%SYSTEM.OBJ,MethodName="Load",Arg1="c:\iris\test.xml",Arg2="ck"
```

**Run a routine and pass two arguments to it; equivalent of `$$Tag2^ZTEST("arg1","arg2")`:**

```
Execute:Namespace="%SYS",RoutineName="$$Tag2^ZTEST",Arg1="arg1",Arg2="arg2"
```

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change `Execute` by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

# CreateComport

---

Create a COM port definition in the [ComPorts] section of the CPF.

## Synopsis

```
[Actions]  
CreateComport:Name=ComPortName[,Additional Properties]
```

## Description

CreateComport defines a COM port in the [ComPorts] section of the CPF. InterSystems IRIS® creates the defined COM port when processing the [\[Actions\]](#) section during a configuration merge. You can specify any possible COM port properties in the CreateComport definition. During the merge, InterSystems IRIS runs Config.ComPorts.Create() to create the COM port defined by CreateComport. It only requires you to define *Name*. The complete list of COM port properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateComport by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Terminal I/O](#) for more information about COM ports.

# DeleteComport

---

Delete a COM port definition in the [ComPorts] section of the CPF.

## Synopsis

```
[Actions]  
DeleteComport:Name=ComPortName
```

## Description

DeleteComport deletes a defined COM port in the [ComPorts] section of the CPF. InterSystems IRIS® deletes the defined COM port when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.ComPorts.Delete() to delete the COM port. It only requires you to define the *Name* parameter to identify the COM port.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateComport by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Terminal I/O](#) for more information about COM ports.

# ModifyComport

---

Modify a COM port definition in the [ComPorts] section of the CPF.

## Synopsis

```
[Actions]  
ModifyComport:Name=ComPortName, [PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyComport modifies a COM port defined in the [ComPorts] section of the CPF. InterSystems IRIS® modifies the defined COM port when processing the [\[Actions\]](#) section during a configuration merge. You specify which COM port properties you want to modify in the ModifyComport definition. During the merge, InterSystems IRIS runs Config.ComPorts.Modify() to modify the COM port. The complete list of COM port properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateComport by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Terminal I/O](#) for more information about COM ports.

---

# ModifyConfig

---

Modify the CPF [Config] parameters.

## Synopsis

```
[Actions]  
ModifyConfig:[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyConfig modifies the [Config] parameters in the CPF. InterSystems IRIS® modifies the defined properties when processing the [\[Actions\]](#) section during a configuration merge. You specify which properties you want to modify in the ModifyConfig definition. During the merge, InterSystems IRIS runs Config.Config.Modify() to modify the properties. The complete list of [config] properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateComport by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [\[config\]](#) for more details about each of the properties you can modify.

---

# CreateDatabase

---

Create a new database.

## Synopsis

```
[Actions]    CreateDatabase:Name=DatabaseName,Directory=DatabaseDirectory[,Additional Properties]
```

## Description

CreateDatabase defines a database. InterSystems IRIS® creates that database when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible database properties in the CreateDatabase definition. During the merge, InterSystems IRIS runs SYS.Database.CreateDatabase() to create the database defined by CreateDatabase. Only the *Name* and *Directory* properties are required; the complete list of properties is available in the class reference.

When using CreateDatabase on an ECP client to add a remote database, there are two unique required properties. These are:

- *Server* – This specifies the name of the ECP server where the remote database is located.
- *LogicalOnly* (1 or 0) – When set to 1, this prevents CreateDatabase from creating a physical database on the ECP client.

The section below contains an example of using these properties to add a remote database.

## Example

Upon running a configuration merge, the following example creates the **Customers** database, specifying the initial size and maximum size:

```
[Actions]
CreateDatabase:Name=Customers,Directory=/IRIS/mgr/Customers,Size=5368,MaxSize=536871
```

If the **Customers** database is a remote database located on **Node1**, the [Actions] section would instead look like:

```
[Actions]
CreateDatabase:Name=Customers,Directory=/IRIS/mgr/Customers,Server=Node1,LogicalOnly=1
```

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateDatabase by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about creating and editing a database from the Management Portal, see [Configuring Databases](#).

---

# DeleteDatabase

---

Delete an existing database.

## Synopsis

```
[Actions]    DeleteDatabase:Name=DatabaseName,Directory=DatabaseDirectory
```

## Description

DeleteDatabase specifies a database to delete. InterSystems IRIS® deletes that database when processing the [\[Actions\]](#) section during a configuration merge.

Specify the *Name* and the *Directory* of the database to delete. During the merge, InterSystems IRIS runs SYS.Database.DeleteDatabase() to delete the database specified by DeleteDatabase.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteDatabase by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about deleting a database from the Management Portal, see [Configuring Databases](#).

# ModifyDatabase

---

Modify an existing database.

## Synopsis

```
[Actions]    ModifyDatabase:Name=DatabaseName,Directory=DatabaseDirectory[,Additional Properties]
```

## Description

ModifyDatabase specifies changes to make to a database. InterSystems IRIS® modifies the specified database when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible database properties in the ModifyDatabase definition. During the merge, InterSystems IRIS runs SYS.Database.ModifyDatabase() to modify the database as specified by ModifyDatabase. The complete list of database properties is available in the class reference.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyDatabase by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about modifying a database from the Management Portal, see [Configuring Databases](#).



---

# CreateDatabaseFile

---

Create a database file. Does not register the database in InterSystems IRIS®.

## Synopsis

```
[Actions]  
CreateDatabaseFile:Directory=DatabaseDirectory[,Additional Properties]
```

## Description

CreateDatabaseFile specifies the location on the host file system where InterSystems IRIS® creates a database file. InterSystems IRIS does not register this database in the instance. InterSystems IRIS creates the database file at the specified location when processing the [\[Actions\]](#) section during a configuration merge. You can specify additional properties for the database file. The complete list of database properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateDatabaseFile by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Configuring Databases](#) for information about databases, .

## DeleteDatabaseFile

---

Delete a database file.

### Synopsis

```
[Actions]  
DeleteDatabaseFile:Directory=DatabaseDirectory
```

### Description

DeleteDatabaseFile specifies the location on the host file system where InterSystems IRIS® deletes a database file. InterSystems IRIS deletes the database file at the specified location when processing the [\[Actions\]](#) section during a configuration merge.

### Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateDatabaseFile by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Configuring Databases](#) for information about databases, .

---

# ModifyDatabaseFile

---

Modify a database file.

## Synopsis

```
[Actions]  
ModifyDatabaseFile:Directory=DatabaseDirectory,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyDatabaseFile specifies the location on the host file system where InterSystems IRIS® modifies a database file. InterSystems IRIS modifies the database file at the specified location according to the defined properties when processing the [\[Actions\]](#) section during a configuration merge. The complete list of database properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateDatabaseFile by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Configuring Databases](#) for information about databases, .

---

# CreateDevice

---

Create a device.

## Synopsis

```
[Action]  
CreateDevice:Name=DeviceName,PhysicalDevice=PhysicalDeviceName,SubType=DeviceSubtypes,Type=x[,Additional  
Properties]
```

Where the value of *x* is one of the following options:

- TRM — Terminal
- SPL — Spooling device
- MT — Magnetic tape drive
- BT — Cartridge tape drive
- IPC — Interprocess communication
- OTH — Any other device including printers and sequential files

## Description

CreateDevice defines a device. InterSystems IRIS® creates that device when processing the [\[Actions\]](#) section during a configuration merge. You can specify any possible device properties in the CreateDevice definition. During the merge, InterSystems IRIS runs Config.Devices.Create() to create the device defined by CreateDevice. Only the *Name*, *PhysicalDevice*, *SubType*, and *Type* properties are required; the complete list of properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateDevice by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Introduction to I/O](#) for more details on devices.

# DeleteDevice

---

Delete a device.

## Synopsis

```
[Actions] DeleteDevice:Name=DeviceName
```

## Description

DeleteDevice deletes an existing device. InterSystems IRIS® deletes the specified device when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.Devices.Delete() to delete the device defined by DeleteDevice.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteDevice by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Introduction to I/O](#) for more details on devices.

# ModifyDevice

---

Modify a device.

## Synopsis

```
[Actions]  
ModifyDevice:Name=DeviceName , [PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyDevice specifies changes to make to an existing device. InterSystems IRIS® modifies this device when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.Devices.Modify() to modify the device defined by ModifyDevice.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyDevice by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Introduction to I/O](#) for more details on devices.

---

# CreateDeviceSubType

---

Create a device subtype.

## Synopsis

```
[Actions] CreateDeviceSubType:Name=DeviceSubTypeName,ScreenLength=n[,Additional Properties]
```

Where *n* is the number of lines that comprise one screen or page for the device. Default is zero.

## Description

CreateDeviceSubType defines a device subtype. InterSystems IRIS® creates that device subtype when processing the [\[Actions\]](#) section during a configuration merge. You can specify any possible device subtype properties in the CreateDeviceSubType definition. During the merge, InterSystems IRIS runs Config.DeviceSubTypes.Create() to create the device subtype defined by CreateDeviceSubType. Only the *Name* and *ScreenLength* properties are required; the complete list of properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateDeviceSubType by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Introduction to I/O](#) for more details on devices.

# DeleteDeviceSubType

---

Delete a device subtype.

## Synopsis

[Actions]  
`DeleteDeviceSubType:Name=DeviceSubTypeName`

## Description

DeleteDeviceSubType deletes an existing device subtype. InterSystems IRIS® deletes the specified device subtype when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs `Config.DeviceSubTypes.Delete()` to delete the device subtype defined by DeleteDeviceSubType.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteDeviceSubType by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Introduction to I/O](#) for more details on devices.



---

# ModifyDeviceSubType

---

Modify a device subtype.

## Synopsis

```
[Actions]  
ModifyDeviceSubType:Name=DeviceSubTypeName , [PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyDeviceSubType specifies changes to make to an existing device subtype. InterSystems IRIS® modifies this device subtype when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.DeviceSubTypes.Modify() to modify the device subtype defined by ModifyDeviceSubType.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyDeviceSubType by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Introduction to I/O](#) for more details on devices.

# CreateDocDB

---

Create a document database (DocDB).

## Synopsis

```
[Actions]  
CreateDocDB:Name=DocDBName,Enabled=n[,Additional Properties]
```

Where *n* is either 0 for disabled or 1 for enabled. Default is 1.

## Description

CreateDocDB defines a document database. InterSystems IRIS® creates that document database when processing the [\[Actions\]](#) section during a configuration merge. You can specify any possible document database properties in the CreateDocDB definition. During the merge, InterSystems IRIS runs Security.DocDBs.Create() to create the document database defined by CreateDocDB. Only the *Name* and *Enabled* properties are required; the complete list of properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateDocDB by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Introduction to Document Database](#) for more details on document databases.

# DeleteDocDB

---

Delete a document database (DocDB).

## Synopsis

```
[Actions]  
DeleteDocDB:Name=DocDBName
```

## Description

DeleteDocDB deletes an existing document database. InterSystems IRIS® deletes the specified document database when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs `Security.DocDBs.Delete()` to delete the document database defined by DeleteDocDB.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteDocDB by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Introduction to Document Database](#) for more details on document databases.

# ModifyDocDB

---

Modify a document database (DocDB).

## Synopsis

```
[Actions]  
ModifyDocDB:Name=DocDBName,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyDocDB specifies changes to make to an existing document database. InterSystems IRIS® modifies this document database when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.DocDBs.Modify() to modify the document database defined by ModifyDocDB.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyDocDB by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Introduction to Document Database](#) for more details on document databases.

# ModifyECP

---

Modify an ECP configuration.

## Synopsis

```
[Actions]  
ModifyECP:Name=ECPConfigName,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyECP specifies changes to make to an existing ECP configuration. InterSystems IRIS® modifies this ECP configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs `Config.ECP.Modify()` to modify the ECP configuration defined by `ModifyECP`.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change `ModifyECP` by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Deploying ECP](#) for more details on ECP configurations.

# CreateECPServer

---

Create an ECP server.

## Synopsis

[Actions]

```
CreateECPServer:Name=ECPServerName,Address=Hostname,MirrorConnection=x,Port=n,SSLConfig=z
```

Where  $x$  is one of the following three values:

- 0 — Non-mirrored connection. Default.
- 1 — Mirrored connection.
- -1 — Mirrored connection restricted to the configured mirror member only.

Where  $n$  is the port number to connect to. Defaults to the superserver port.

Where  $z$  is either 0 for do not use the TLS configuration (%ECPClient) or 1 for do use the TLS configuration (%ECPClient). Default is 0.

## Description

CreateECPServer defines an ECP server. InterSystems IRIS® creates that ECP server when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.ECPServers.Create() to create the ECP server defined by CreateECPServer. All properties are required; the complete list of properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateECPServer by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Deploying ECP](#) for more details on ECP configurations.

# DeleteECPServer

---

Delete an ECP server.

## Synopsis

```
[Actions]  
DeleteECPServer:Name=ECPServerName
```

## Description

DeleteECPServer deletes an existing ECP server. InterSystems IRIS® deletes the specified ECP server when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.ECPServers.Delete() to delete the ECP server defined by DeleteECPServer.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteECPServer by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Deploying ECP](#) for more details on ECP configurations.

# ModifyECPServer

---

Modify an ECP server.

## Synopsis

```
[Actions]  
ModifyECPServer:Name=ECPServerName,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyECPServer specifies changes to make to an existing ECP server. InterSystems IRIS® modifies this ECP server when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.ECPServers.Modify() to modify the ECP server defined by ModifyECPServer.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyECPServer by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Deploying ECP](#) for more details on ECP configurations.



---

# CreateEvent

---

Create a new audit event.

## Synopsis

```
[Actions] CreateEvent:Source=EventSource,Type=EventType,Name=EventName,Enabled=x[,Additional Properties]
```

Where  $x$  is 1 for enabled or 0 for disabled.

## Description

CreateEvent specifies an audit event to create. InterSystems IRIS® creates that audit event when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible audit event properties in the CreateEvent definition. During the merge, InterSystems IRIS runs Security.Events.Create() to create the audit event defined by CreateEvent.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateEvent by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about creating an audit event from the Management Portal, see [Auditing](#).

# DeleteEvent

---

Delete an audit event.

## Synopsis

[Actions]    DeleteEvent:Source=*EventSource*,Type=*EventType*,Name=*EventName*

## Description

DeleteEvent deletes an existing audit event. InterSystems IRIS® deletes the specified audit event when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.Events.Delete() to delete the audit event defined by DeleteEvent.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteEvent by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about audit events, see [Auditing](#).

---

# ModifyEvent

---

Modify an audit event.

## Synopsis

```
[Actions]    ModifyEvent:Source=EventSource,Type=EventType,Name=EventName,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyEvent specifies changes to make to an existing audit event. InterSystems IRIS® modifies this audit event when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.Events.Modify() to modify the audit event defined by ModifyEvent.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyEvent by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about audit events, see [Auditing](#).

# ModifyJournal

---

Modify the [Journal] CPF settings.

## Synopsis

```
[Actions]  
ModifyJournal:[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyJournal specifies changes to make to the journal settings. InterSystems IRIS® modifies the journal settings when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.Journal.Modify() to modify the journal settings defined by ModifyJournal.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyJournal by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about journaling, see [Journaling](#).

# CreateLDAPConfig

---

Create an LDAP configuration.

## Synopsis

```
[Actions]  
CreateLDAPConfig:Name=LDAPConfigName[,Additional Properties]
```

## Description

CreateLDAPConfig defines an LDAP configuration. InterSystems IRIS® creates that LDAP configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.LDAPConfigs.Create() to create the LDAP configuration defined by CreateLDAPConfig. Many properties are required; the complete list of properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateLDAPConfig by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [LDAP Configuration Fields](#) for more information about LDAP configurations.

## DeleteLDAPConfig

---

Delete an LDAP configuration.

### Synopsis

```
[Actions]  
DeleteLDAPConfig:Name=LDAPConfigName
```

### Description

DeleteLDAPConfig deletes an existing LDAP configuration. InterSystems IRIS® deletes the specified LDAP configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.LDAPConfigs.Delete() to delete the LDAP configuration defined by DeleteLDAPConfig.

### Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteLDAPConfig by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [LDAP Configuration Fields](#) for more information about LDAP configurations.

---

# ModifyLDAPConfig

---

Modify an LDAP configuration.

## Synopsis

```
[Actions]  
ModifyLDAPConfig:Name=LDAPConfigName, [PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyLDAPConfig specifies changes to make to an existing LDAP configuration. InterSystems IRIS® modifies the LDAP configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.LDAPConfigs.Modify() to modify the LDAP configuration defined by ModifyLDAPConfig.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyLDAPConfig by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [LDAP Configuration Fields](#) for more information about LDAP configurations.

# CreateLicenseServer

---

Create a license server.

## Synopsis

[Actions]

CreateLicenseServer:Name=*LicenseServerName*,Address=*LicenseServerAddress*,Port=*n*[*Additional Properties*]

## Description

CreateLicenseServer defines a license server. InterSystems IRIS® creates that license server when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.LicenseServers.Create() to create the license server defined by CreateLicenseServer. The *Name*, *Address* and *Port* properties are required; the complete list of properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateLicenseServer by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Configuring a License Server](#) for more information about license servers.



# DeleteLicenseServer

---

Delete a license server.

## Synopsis

```
[Actions]  
DeleteLicenseServer:Name=LicenseServerName
```

## Description

DeleteLicenseServer deletes an existing license server. InterSystems IRIS® deletes the specified license server when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.LicenseServers.Delete() to delete the license server defined by DeleteLicenseServer.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteLicenseServer by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Configuring a License Server](#) for more information about license servers.

# ModifyLicenseServer

---

Modify a license server.

## Synopsis

[Actions]

```
ModifyLicenseServer:Name=LicenseServerName , [PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyLicenseServer specifies changes to make to an existing license server. InterSystems IRIS® modifies the license server when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.LicenseServers.Modify() to modify the license server defined by ModifyLicenseServer.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyLicenseServer by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Configuring a License Server](#) for more information about license servers.

# CreateMagTapes

---

Create a magnetic tape definition in the [MagTapes] section of the CPF.

## Synopsis

```
[Actions]  
CreateMagTapes:Name=MagTapeName,SystemDevice=TapeDeviceName
```

## Description

CreateMagTapes defines a magnetic tape. InterSystems IRIS® creates that magnetic tape definition when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.MagTapes.Create() to create the magnetic tape definition defined by CreateMagTapes. The *Name* and *SystemDevice* properties are required; the complete list of properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateMagTapes by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

## DeleteMagTapes

---

Delete a magnetic tape definition in the [MagTapes] section of the CPF.

### Synopsis

```
[Actions]  
DeleteMagTapes:Name=MagTapeName
```

### Description

DeleteMagTapes deletes an existing magnetic tape definition. InterSystems IRIS® deletes the specified magnetic tape definition when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.MagTapes.Delete() to delete the magnetic tape definition defined by DeleteMagTapes.

### Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteMagTapes by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

# ModifyMagTapes

---

Modify a magnetic tape definition in the [MagTapes] section of the CPF.

## Synopsis

```
[Actions]  
ModifyMagTapes:Name=MagTapeName,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyMagTapes specifies changes to make to an existing magnetic tape definition. InterSystems IRIS® modifies the magnetic tape definition when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.MagTapes.Modify() to modify the magnetic tape definition defined by ModifyMagTapes.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyMagTapes by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

# CreateMapGlobal

---

Create a new global mapping.

## Synopsis

```
[Actions]    CreateMapGlobal:Namespace=Namespace,Name=MappingName,Database=Database[,Additional Properties]
```

## Description

CreateMapGlobal specifies a global mapping to create. InterSystems IRIS® creates that global mapping when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible global mapping properties in the CreateMapGlobal definition. During the merge, InterSystems IRIS runs Config.MapGlobals.Create() to create the global mapping defined by CreateMapGlobal.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateMapGlobal by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about creating a global mapping from the Management Portal, see [Global Mappings](#).

---

# DeleteMapGlobal

---

Delete an existing global mapping.

## Synopsis

```
[Actions]    DeleteMapGlobal:Namespace=Namespace,Name=MappingName,Database=Database
```

## Description

DeleteMapGlobal specifies a global mapping to delete. InterSystems IRIS® deletes the specified global mapping when processing the [\[Actions\]](#) section during a configuration merge.

Specify the *Namespace*, *Name*, and *Database* for the global mapping to delete. During the merge, InterSystems IRIS runs `Config.MapGlobals.Delete()` to delete the global mapping defined by DeleteMapGlobal.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteMapGlobal by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

# ModifyMapGlobal

---

Modify an existing global mapping.

## Synopsis

```
[Actions]    ModifyMapGlobal:Namespace=Namespace,Name=MappingName,Database=Database[,Additional Properties]
```

## Description

ModifyMapGlobal specifies changes to make to a global mapping. InterSystems IRIS® modifies the specified global mapping when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible global mapping properties in the ModifyMapGlobal definition. During the merge, InterSystems IRIS runs Config.MapGlobals.Modify() to make the changes defined by ModifyMapGlobal.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyMapGlobal by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).



# CreateMapPackage

---

Create a package mapping.

## Synopsis

```
[Actions]  
CreateMapPackage:Namespace=NSToMap, Name=MappingName, Database=DBToMap
```

## Description

CreateMapPackage defines a package mapping. InterSystems IRIS® creates that package mapping when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.MapPackages.Create() to create the package mapping defined by CreateMapPackage. The *Namespace*, *Name*, and *Database* properties are required; the complete list of properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateMapPackage by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Adding Mappings to a Namespace](#) for more information on mappings.

# DeleteMapPackage

---

Delete a package mapping.

## Synopsis

[Actions]  
`DeleteMapPackage:Namespace=NSToMap, Name=MappingName`

## Description

DeleteMapPackage deletes an existing package mapping. InterSystems IRIS® deletes the specified package mapping when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs `Config.Map-Packages.Delete()` to delete the package mapping defined by DeleteMapPackage.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteMapPackage by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Adding Mappings to a Namespace](#) for more information on mappings.

# ModifyMapPackage

---

Modify a package mapping.

## Synopsis

```
[Actions]  
ModifyMapPackage:Namespace=NSToMap,Name=MappingName,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyMapPackage specifies changes to make to an existing package mapping. InterSystems IRIS® modifies the package mapping when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.MapPackages.Modify() to modify the package mapping defined by ModifyMapPackage.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyMapPackage by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Adding Mappings to a Namespace](#) for more information on mappings.

# CreateMapRoutine

---

Create a new routine mapping.

## Synopsis

```
[Actions] CreateMapRoutine:Namespace=Namespace,Name=MappingName,Database=Database[,Additional Properties]
```

## Description

CreateMapRoutine specifies a routine mapping to create. InterSystems IRIS® creates that routine mapping when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible routine mapping properties in the CreateMapRoutine definition. During the merge, InterSystems IRIS runs Config.MapRoutines.Create() to create the routine mapping defined by CreateMapRoutine.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateMapRoutine by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about creating a routine mapping from the Management Portal, see [Routine Mappings](#).

---

# DeleteMapRoutine

---

Delete an existing routine mapping.

## Synopsis

```
[Actions]    DeleteMapRoutine:Namespace=Namespace,Name=MappingName,Database=Database[,Additional  
Properties]
```

## Description

DeleteMapRoutine specifies a routine mapping to delete. InterSystems IRIS® deletes the specified routine mapping when processing the [\[Actions\]](#) section during a configuration merge.

Specify the *Namespace*, *Name*, and *Database* for the routine mapping to delete. During the merge, InterSystems IRIS runs `Config.MapRoutines.Delete()` to delete the routine mapping defined by DeleteMapRoutine.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteMapRoutine by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

# ModifyMapRoutine

---

Modify an existing routine mapping.

## Synopsis

```
[Actions]    ModifyMapRoutine:Namespace=Namespace,Name=MappingName,Database=Database[,Additional Properties]
```

## Description

ModifyMapRoutine specifies changes to make to a routine mapping. InterSystems IRIS® modifies the specified routine mapping when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible routine mapping properties in the ModifyMapRoutine definition. During the merge, InterSystems IRIS runs Config.MapRoutines.Modify() to make the changes defined by ModifyMapRoutine.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyMapRoutine by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

---

# ConfigMirror

---

Configure a mirrored deployment.

## Synopsis

```
[Actions]  
ConfigMirror:Name=MirrorName[,Additional Properties]
```

## Description

ConfigMirror defines a mirror configuration. InterSystems IRIS® deploys that mirror configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs a series of API calls to deploy the mirror configuration. See [ConfigMirror Arguments](#) for details the arguments available for this action and their usage.

ConfigMirror is reentrant and will exit if the mirror already exists without updating any properties. The exception is the property `ArbiterURL` which allows for the addition of an arbiter if one was not previously configured.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ConfigMirror by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Configuring Mirroring](#) for more information about mirror configurations.

# CreateMirror

---

Create a mirror configuration.

## Synopsis

```
[Actions]  
CreateMirror:Name=MirrorName[,Additional Properties]
```

## Description

CreateMirror defines a mirror configuration. InterSystems IRIS® creates that mirror configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.Mirrors.Create() to create the mirror configuration defined by CreateMirror. The *Name* property is required; the complete list of properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateMirror by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Configuring Mirroring](#) for more information about mirror configurations.



---

# DeleteMirror

---

Delete a mirror configuration.

## Synopsis

```
[Actions]  
DeleteMirror:Name=MirrorName
```

## Description

DeleteMirror deletes an existing mirror configuration. InterSystems IRIS® deletes the specified mirror configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.Mirrors.Delete() to delete the mirror configuration defined by DeleteMirror.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteMirror by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Configuring Mirroring](#) for more information about mirror configurations.

# ModifyMirror

---

Modify a mirror configuration.

## Synopsis

```
[Actions]  
ModifyMirror:Name=MirrorName,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyMirror specifies changes to make to an existing mirror configuration. InterSystems IRIS® modifies the mirror configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.Mirrors.Modify() to modify the mirror configuration defined by ModifyMirror.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyMirror by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Configuring Mirroring](#) for more information about mirror configurations.

# ModifyMirrorMember

---

Modify a mirror member.

## Synopsis

```
[Actions]  
ModifyMirrorMember:Name=MirrorMemberName,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyMirrorMember specifies changes to make to an existing mirror member configuration. InterSystems IRIS® modifies the mirror member configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.MirrorMember.Modify() to modify the mirror member configuration defined by ModifyMirrorMember.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyMirrorMember by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Configuring Mirroring](#) for more information about mirror configurations.

# ModifyMiscellaneous

---

Modify the [Miscellaneous] section of the CPF.

## Synopsis

```
[Actions]
ModifyMiscellaneous:[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyMiscellaneous specifies changes to make to the [Miscellaneous] section of the CPF. InterSystems IRIS® modifies the [Miscellaneous] section of the CPF when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.Miscellaneous.Modify() to modify the [Miscellaneous] section of the CPF defined by ModifyMiscellaneous.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyMiscellaneous by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

# ModifyMonitor

---

Modify the [Monitor] section of the CPF.

## Synopsis

```
[Actions]  
ModifyMonitor:[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyMonitor specifies changes to make to the [Monitor] section of the CPF. InterSystems IRIS® modifies the [Monitor] section of the CPF when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.Monitor.Modify() to modify the [Monitor] section of the CPF defined by ModifyMonitor.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyMiscellaneous by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

---

# CreateNamespace

---

Create a new namespace.

## Synopsis

[Actions]

CreateNamespace:Name=NamespaceName,Globals=DatabaseName[,Routines=DatabaseName,TempGlobals=DatabaseName]

## Description

CreateNamespace defines a namespace. InterSystems IRIS® creates that namespace when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible namespace properties in the CreateNamespace definition. During the merge, InterSystems IRIS runs Config.Namespaces.Create() to create the namespace defined by CreateNamespace. The namespace properties are:

- *Name* (required) – The name for the new namespace.
- *Globals* (required) – The default globals database for the new namespace.
- *Routines* – The default routines database for the new namespace. If unspecified, this uses the same database as *Globals*.
- *TempGlobals* – The default temporary globals database for the new namespace. If unspecified, this uses the **IRISTEMP** database.

## Example

Upon running a configuration merge, the following example creates the **Sales** namespace, which uses the **Sales** database for globals and the **SYS** database for routines.

[Actions]

CreateNamespace:Name=Sales,Globals=Sales,Routines=SYS

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateNamespace by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about creating and editing a namespace from the Management Portal, see [Configuring Namespaces](#).

# DeleteNamespace

---

Delete an existing namespace.

## Synopsis

```
[Actions]    DeleteNamespace:Name=NamespaceName
```

## Description

DeleteNamespace specifies a namespace to delete. InterSystems IRIS® deletes that namespace when processing the [\[Actions\]](#) section during a configuration merge.

Specify the *Name* of the namespace to delete. During the merge, InterSystems IRIS runs Config.Namespaces.Delete() to delete the namespace specified by DeleteNamespace.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteNamespace by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about deleting a namespace from the Management Portal, see [Configuring Namespaces](#).

# ModifyNamespace

---

Modify an existing namespace.

## Synopsis

```
[Actions]    ModifyNamespace:Name=NamespaceName[Additional Properties]
```

## Description

ModifyNamespace specifies changes to make to a namespace. InterSystems IRIS® modifies the specified namespace when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible namespace properties in the ModifyNamespace definition. During the merge, InterSystems IRIS runs Config.Namespaces.Modify() to modify the namespace as specified by ModifyNamespace. The complete list of namespace properties is available in the class reference.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyNamespace by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about modifying a namespace from the Management Portal, see [Configuring Namespaces](#).



---

# CreateResource

---

Create a new resource.

## Synopsis

```
[Actions]    CreateResource:Name=ResourceName,PublicPermission=[R/W/U]
```

## Description

CreateResource specifies a resource to create. InterSystems IRIS® creates that resource when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible resource properties in the CreateResource definition. During the merge, InterSystems IRIS runs Security.Resources.Create() to create the resource defined by CreateResource. The complete list of resource properties is available in the class reference.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateResource by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about creating resources from the Management Portal, see [Create or Edit a Resource](#).

# DeleteResource

---

Delete an existing resource.

## Synopsis

[Actions]    DeleteResource:Name=*ResourceName*

## Description

DeleteResource specifies a resource to delete. InterSystems IRIS® deletes that resource when processing the [\[Actions\]](#) section during a configuration merge.

Specify the *Name* of the resource to delete. During the merge, InterSystems IRIS runs Security.Resources.Delete() to delete the resource specified by DeleteResource.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteResource by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

You can also delete resources from the **Resources** page of the Management Portal (**System Administration** > **Security** > **Resources**).

---

# ModifyResource

---

Modify an existing resource.

## Synopsis

```
[Actions]    ModifyResource:Name=ResourceName[,Additional Properties]
```

## Description

ModifyResource specifies changes to make to a resource. InterSystems IRIS® modifies the specified resource when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible resource properties in the ModifyResource definition. During the merge, InterSystems IRIS runs Security.Resources.Modify() to make the changes defined by ModifyResource. The complete list of resource properties is available in the class reference.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyResource by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about modifying a resource from the Management Portal, see [Create or Edit a Resource](#).

# CreateRole

---

Create a new role.

## Synopsis

```
[Actions]    CreateRole:Name=RoleName,Resources=ResourceName(s)[,Additional Properties]
```

## Description

CreateRole defines a role. InterSystems IRIS® creates that role when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible role properties in the CreateRole definition. During the merge, InterSystems IRIS runs Security.Roles.Create() to create the role defined by CreateRole. The complete list of role properties is available in the class reference.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateRole by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about creating roles from the Management Portal, see [Create Roles](#).

# DeleteRole

---

Delete an existing role.

## Synopsis

```
[Actions]    DeleteRole:Name=RoleName
```

## Description

DeleteRole specifies a role to delete. InterSystems IRIS® deletes that role when processing the [\[Actions\]](#) section during a configuration merge.

Specify the *Name* of the role to delete. During the merge, InterSystems IRIS runs Security.Roles.Delete() to delete the role specified by DeleteRole.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteRole by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about deleting a role from the Management Portal, see [Delete a Role](#).

# ModifyRole

---

Modify an existing role.

## Synopsis

```
[Actions]    ModifyRole:Name=RoleName[,Additional Properties]
```

## Description

ModifyRole specifies changes to make to a role. InterSystems IRIS® modifies the specified role when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible role properties in the ModifyRole definition. During the merge, InterSystems IRIS runs Security.Roles.Modify() to make the changes defined by ModifyRole. The complete list of SPECIFIC properties is available in the class reference.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyRole by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about modifying a role from the Management Portal, see [Manage Roles](#).

---

# ModifyService

---

Modify a service.

## Synopsis

```
[Actions]  
ModifyService:Name=ServiceName,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyService specifies changes to make to an existing service. InterSystems IRIS® modifies the existing service when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.Services.Modify() to modify the existing service defined by ModifyService.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyService by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Services](#) for more information on services.

# ConfigShardedCluster

---

Configure a sharded cluster.

## Synopsis

```
[Actions]  
ConfigShardedCluster:Name=ClusterName, [PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ConfigShardedCluster specifies changes to make to a sharded cluster. InterSystems IRIS® modifies the sharded cluster when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs a sequence of API calls to modify the sharded cluster defined by ConfigShardedCluster.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ConfigShardedCluster by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Sharding](#) for more information on sharding configurations.



# GrantPrivilege

---

Grant an object privilege to a user.

## Synopsis

```
[Actions]  
GrantPrivilege:ObjPriv=a,ObjList=b,Type=c,User=d
```

Where *a* is a comma delimited string of actions to grant or \* for all actions.

Where *b* is a comma delimited list of SQL object names or \* for all objects.

Where *c* is Table, View, Schema, Stored Procedures, or ML Configuration.

Where *d* is a comma delimited list of users.

## Description

GrantPrivilege lets you grant an object privilege to users via this call instead of using the SQL GRANT statement. This does not include grant privileges. InterSystems IRIS® grants the privileges when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs %SYSTEM.SQL.Security.GrantPrivilege() to grant the object privileges defined by GrantPrivilege.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change GrantPrivilege by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [GRANT \(SQL\)](#) for information on the SQL command.

# RevokePrivilege

---

Revoke an object privilege from a user.

## Synopsis

```
[Actions]  
RevokePrivilege:ObjPriv=a,ObjList=b,Type=c,User=d[ ,AdditionalProperties]
```

Where *a* is a comma delimited string of actions to revoke or \* for all actions.

Where *b* is a comma delimited list of SQL object names or \* for all objects.

Where *c* is Table, View, Schema, Stored Procedures, or ML Configuration.

Where *d* is a comma delimited list of users.

## Description

RevokePrivilege lets you revoke an object privilege to users via this call instead of using the SQL REVOKE statement. InterSystems IRIS® revokes the privileges when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs %SYSTEM.SQL.Security.RevokePrivilege() to revoke the object privileges defined by RevokePrivilege.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change RevokePrivilege by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Revoke \(SQL\)](#) for information on the SQL command.

---

# ModifySQL

---

Modify the [SQL] section of the CPF.

## Synopsis

```
[Actions]  
ModifySQL:[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifySQL specifies changes to make to the SQL settings. InterSystems IRIS® modifies the SQL settings when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.SQL.Modify() to modify the SQL settings defined by ModifySQL.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifySQL by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [\[SQL\]](#) for information on the [SQL] CPF section.

# CreateSqlSysDatatype

---

Create a SQL system datatype definition in the [SqlSysDatatypes] section of the CPF.

## Synopsis

```
[Actions]
CreateSqlSysDatatype:Name=DatatypeName,Datatype=datatype
```

## Description

CreateSqlSysDatatype defines a SQL system datatype. InterSystems IRIS® creates that SQL system datatype when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.SqlSys-Datatypes.Create() to create the SQL system datatype defined by CreateSqlSysDatatype. The *Name* and *Datatype* properties are required; the complete list of properties is available in the class reference.

## Example

The following example demonstrates how to use the CreateSqlSysDatatype operation in the configuration merge file:

```
[Actions]
CreateSqlSysDatatype:Name=BigInt,Datatype=%Library.BigInt
```

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateSqlSysDatatype by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Datatypes \(SQL\)](#) for more information on SQL datatypes.

# DeleteSqlSysDatatype

---

Delete a SQL system datatype definition in the [SqlSysDatatypes] section of the CPF.

## Synopsis

```
[Actions]  
DeleteSqlSysDatatype:Name=DatatypeName
```

## Description

DeleteSqlSysDatatype deletes an existing SQL system datatype. InterSystems IRIS® deletes the specified SQL system datatype when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.SqlSysDatatypes.Delete() to delete the SQL system datatype defined by DeleteSqlSysDatatype.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteSqlSysDatatype by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Datatypes \(SQL\)](#) for more information on SQL datatypes.

# ModifySqlSysDatatype

---

Modify a SQL system datatype definition in the [SqlSysDatatypes] section of the CPF.

## Synopsis

```
[Actions]  
ModifySqlSysDatatype:Name=DatatypeName,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifySqlSysDatatype specifies changes to make to an existing SQL system datatype. InterSystems IRIS® modifies the SQL system datatype when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.SqlSysDatatypes.Modify() to modify the SQL system datatype defined by ModifySqlSysDatatype.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifySqlSysDatatype by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Datatypes \(SQL\)](#) for more information on SQL datatypes.

# CreateSqlUserDatatype

Create a SQL user datatype definition in the [SqlUserDatatypes] section of the CPF.

## Synopsis

```
[Actions]  
CreateSqlUserDatatype:Name=Name,Datatype=datatype
```

## Description

CreateSqlUserDatatype defines a SQL user datatype. InterSystems IRIS® creates that SQL user datatype when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.SqlUserDatatypes.Create() to create the SQL user datatype defined by CreateSqlUserDatatype. The *Name* and *Datatype* properties are required; the complete list of properties is available in the class reference.

## Example

The following example demonstrates how to use the CreateSqlUserDatatype operation in the configuration merge file:

```
[Actions]  
CreateSqlUserDatatype:Name=BigNEWInt,Datatype=%Library.BigNEWInt
```

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateSqlUserDatatype by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Datatypes \(SQL\)](#) for more information on SQL datatypes.

## DeleteSqlUserDatatype

---

Delete a SQL user datatype definition in the [SqlUserDatatypes] section of the CPF.

### Synopsis

```
[Actions]  
DeleteSqlUserDatatype:Name=Name
```

### Description

DeleteSqlUserDatatype deletes an existing SQL user datatype. InterSystems IRIS® deletes the specified SQL user datatype when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.SqlUserDatatypes.Delete() to delete the SQL user datatype defined by DeleteSqlUserDatatype.

### Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteSqlUserDatatype by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Datatypes \(SQL\)](#) for more information on SQL datatypes.



---

# ModifySqlUserDatatype

---

Modify a SQL user datatype definition in the [SqlUserDatatypes] section of the CPF.

## Synopsis

```
[Actions]  
ModifySqlUserDatatype:Name=DatatypeName,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifySqlUserDatatype specifies changes to make to an existing SQL user datatype. InterSystems IRIS® modifies the SQL user datatype when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.SqlUserDatatypes.Modify() to modify the SQL user datatype defined by ModifySqlUserDatatype.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifySqlUserDatatype by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Datatypes \(SQL\)](#) for more information on SQL datatypes.

# CreateSSLConfig

---

Create a TLS configuration.

## Synopsis

```
[Actions]  
CreateSSLConfig:Name=SSLConfigName[,AdditionalProperties]
```

## Description

CreateSSLConfig defines a TLS configuration. InterSystems IRIS® creates that TLS configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.SSLConfigs.Create() to create the TLS configuration defined by CreateSSLConfig. The *Name* property is required; the complete list of properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateSSLConfig by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [About TLS Configurations](#) for more information on TLS configurations.

---

# DeleteSSLConfig

---

Delete a TLS configuration.

## Synopsis

```
[Actions]  
DeleteSSLConfig:Name=SSLConfigName
```

## Description

DeleteSSLConfig deletes an existing TLS configuration. InterSystems IRIS® deletes the specified TLS configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.SSLConfigs.Delete() to delete the TLS configuration defined by DeleteSSLConfig.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteSSLConfig by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [About TLS Configurations](#) for more information on TLS configurations.

# ModifySSLConfig

---

Modify a TLS configuration.

## Synopsis

```
[Actions]  
ModifySSLConfig:Name=SSLConfigName,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifySSLConfig specifies changes to make to an existing TLS configuration. InterSystems IRIS® modifies the TLS configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.SSLConfigs.Modify() to modify the TLS configuration defined by ModifySSLConfig.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifySSLConfig by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [About TLS Configurations](#) for more information on TLS configurations.

---

# ModifyStartup

---

Modify the [Startup] section of the CPF.

## Synopsis

```
[Actions]  
ModifyStartup:[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyStartup specifies changes to make to the startup settings. InterSystems IRIS® modifies the startup settings when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.Startup.Modify() to modify the startup settings defined by ModifyStartup.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyStartup by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Memory and Startup Settings](#) for more information on startup settings.

# CreateServer

---

Create a superserver configuration.

## Synopsis

```
[Actions]  
CreateServer:Port=n,BindAddress=bindaddress[,AdditionalProperties]
```

Where *n* is an open port number on the system.

## Description

CreateServer defines a superserver configuration. InterSystems IRIS® creates that superserver configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.Servers.Create() to create the superserver configuration defined by CreateServer. The *Port* and *BindAddress* properties are required; the complete list of properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateServer by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Managing Superservers](#) for more information on superserver configurations.

---

# DeleteServer

---

Delete a superserver configuration.

## Synopsis

```
[Actions]  
DeleteServer:Port=n,BindAddress=bindaddress
```

Where *n* is the existing superserver's port number on the system.

## Description

DeleteServer deletes an existing superserver configuration. InterSystems IRIS® deletes the specified superserver configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.Servers.Delete() to delete the superserver configuration defined by DeleteServer. The *Port* and *BindAddress* properties are required to identify the superserver.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteServer by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Managing Superservers](#) for more information on superserver configurations.

# ModifyServer

---

Modify a superserver configuration.

## Synopsis

[Actions]

`ModifyServer:Port=n,BindAddress=bindaddress,[PropertiesToModify]`

Where *n* is the existing superserver's port number on the system.

Where you substitute [*PropertiesToModify*] for the properties you want to modify.

## Description

ModifyServer specifies changes to make to an existing superserver configuration. InterSystems IRIS® modifies the superserver configuration when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs `Security.Servers.Modify()` to modify the superserver configuration defined by ModifyServer.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyServer by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Managing Superservers](#) for more information on superserver configurations.



---

# ModifySystem

---

Modify system security settings.

## Synopsis

```
[Actions]  
ModifySystem:Name=Name,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifySystem specifies changes to make to the system security settings. InterSystems IRIS® modifies the system security settings when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Security.System.Modify() to modify the system security settings defined by ModifySystem.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifySystem by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [System-Wide Security Parameters](#) for more information on system security settings.

# CreateUser

---

Create a new user account.

## Synopsis

```
[Actions]    CreateUser:Name=UserName,Roles=Roles[,Additional Properties]
```

## Description

CreateUser defines a user account. InterSystems IRIS® creates that user account when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible user account properties in the CreateUser definition. During the merge, InterSystems IRIS runs Security.Users.Create() to create the account defined by CreateUser. The complete list of user account properties is available in the class reference.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateUser by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about creating user accounts from the Management Portal, see [Create a New User Account](#).

# DeleteUser

---

Delete an existing user account.

## Synopsis

```
[Actions]    DeleteUser:Name=UserName
```

## Description

DeleteUser specifies a user account to delete. InterSystems IRIS® deletes that account when processing the [\[Actions\]](#) section during a configuration merge.

Specify the *Name* of the user to delete. During the merge, InterSystems IRIS runs Security.Users.Delete() to delete the user specified by DeleteUser.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteUser by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about deleting user accounts from the Management Portal, see [Delete a User Account](#).

# ModifyUser

---

Modify an existing user account.

## Synopsis

```
[Actions]    ModifyUser:Name=UserName[,Additional Properties]
```

## Description

ModifyUser specifies changes to make to a user account. InterSystems IRIS® modifies the specified user account when processing the [\[Actions\]](#) section during a configuration merge.

You can specify any possible user properties in the ModifyUser definition. During the merge, InterSystems IRIS runs Security.Users.Modify() to make the changes defined by ModifyUser.

## Changing This Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyUser by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

For information about modifying user accounts from the Management Portal, see [Manage User Accounts](#).

# CreateWorkQueue

---

Create a work queue definition in the [WorkQueues] section of the CPF.

## Synopsis

```
[Actions]  
CreateWorkQueue:Name=WQName,MaxActiveWorkers=n
```

Where  $n$  is a positive integer ( $\geq 0$ ) representing the maximum number of active worker jobs kept in the pool of jobs servicing requests. Default is 0 which represents a dynamic maximum.

## Description

CreateWorkQueue defines a work queue. InterSystems IRIS® creates that work queue when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.WorkQueues.Create() to create the work queue defined by CreateWorkQueue. The *Name* and *MaxActiveWorkers* properties are required; the complete list of properties is available in the class reference.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change CreateWorkQueue by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Introduction to the Work Queue Manager](#) for more information on work queues.

## DeleteWorkQueue

---

Delete a work queue definition in the [WorkQueues] section of the CPF.

### Synopsis

```
[Actions]  
DeleteWorkQueue:Name=WQName
```

### Description

DeleteWorkQueue deletes an existing work queue. InterSystems IRIS® deletes the specified work queue when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs Config.WorkQueues.Delete() to delete the work queue defined by DeleteWorkQueue.

### Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change DeleteWorkQueue by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Introduction to the Work Queue Manager](#) for more information on work queues.

---

# ModifyWorkQueue

---

Modify a work queue definition in the [WorkQueues] section of the CPF.

## Synopsis

```
[Actions]  
ModifyWorkQueue:Name=WQName,[PropertiesToModify]
```

Where you substitute *[PropertiesToModify]* for the properties you want to modify.

## Description

ModifyWorkQueue specifies changes to make to a work queue. InterSystems IRIS® modifies the work queue when processing the [\[Actions\]](#) section during a configuration merge. During the merge, InterSystems IRIS runs `Config.WorkQueues.Modify()` to modify the work queue defined by ModifyWorkQueue.

## Changing this Operation

This operation is designed to be used during a [configuration merge](#). You can change ModifyWorkQueue by editing the merge file in a text editor (as described in [Editing the Active CPF](#)).

See [Introduction to the Work Queue Manager](#) for more information on work queues.





# [Archives]

The configuration parameter file may include an [Archives] section, which defines optional [journal archive targets](#) for use in [configuring journal settings](#).

# Archives

Define a journal archive target for use in [configuring journal settings](#).

## Synopsis

```
[Archives]      Name=Type,Location
```

*Name*, *Type*, and *Location* are strings.

## Description

The `[Archives]` section of the configuration parameter (CPF) file contains an entry for every journal archive target defined for this instance, as follows:

- *Name* is the name of the archive target.
- *Type* is the type of the target, one of the following:
  - `s3` specifies an AWS S3 location.
  - `rsync` specifies an on-premises location that uses either a UNIX® or a Windows directory specification. (Note that if the location is a Windows directory, the system actually uses robocopy rather than rsync when it copies the files.)
- *Location* specifies the actual directory. The format depends on *Type*.
  - For `s3`, **Location** has the following format:

```
s3://directoryname/
```

For example:

```
s3://test-dir/journal-archives/
```

- For `rsync`, **Location** can any of the following formats:

- For UNIX®:

```
/directoryname/
```

Or:

```
server:/directoryname/
```

- For Windows:

```
\\server\share\directoryname\
```

Or:

```
drive:\directoryname\
```

For more information, see [Configuring Journal Archive Targets](#) and [Configuring Journal Settings](#).

## Changing This Parameter

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



# [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.

### Synopsis

[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* can be multiple digits.

The parameters within COMn apply to Windows platforms only.

Each COMn entry provides two comma-separated values that define the following default settings for COM port number *n*:

- *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 Position	Description
Byte 1: Modem Control	<ul style="list-style-type: none"> <li>– '1' Use modem control</li> <li>– '0' Do not use modem control</li> <li>– ' ' No change to modem control (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</li> <li>– ' ' No change to bit size (default)</li> </ul>
Byte 3: Parity	<ul style="list-style-type: none"> <li>– '0' No parity</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 (default)</li> </ul>
Byte 4: Stop Bits	<ul style="list-style-type: none"> <li>– '1' 1 stop bit</li> <li>– '5' 1.5 stop bits</li> <li>– '2' 2 stop bits</li> <li>– ' ' No change to the stop bit setting (default)</li> </ul>
Byte 5: Flow Control	<ul style="list-style-type: none"> <li>– 'X' Use Xon/Xoff flow control</li> <li>– 'C' Use CTS/RTS flow control</li> <li>– 'D' Use DSR/DTR flow control</li> <li>– ' ' No change to flow control (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)</li> <li>– ' ' No change to the DTR state (default)</li> </ul>
Byte 7: \$ZA Error Reporting	<ul style="list-style-type: none"> <li>– '0' Disable \$ZA error reporting</li> <li>– '1' Enable \$ZA error reporting</li> <li>– ' ' No change to \$ZA error reporting (this is the default)</li> </ul>

- *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 comma 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 in the following ways:

- Edit the CPF in a text editor (as described in [Editing the Active CPF](#)).
- Use the Config.ComPorts class.



# [config]

This topic describes the **Advanced Memory Settings** parameters found in the [config] section of the CPF.

# LibPath

---

Add directories to the LD\_LIBRARY\_PATH environment variable. UNIX® systems only.

## Synopsis

```
[config]    LibPath=directory:directory2:directory3[...]
```

*directory* is the full path to a valid directory. The maximum length for LibPath is 1024 characters. By default, no directories are listed.

## Description

LibPath is used for UNIX® systems only. This parameter adds a list of directories to the LD\_LIBRARY\_PATH environment variable (DYLD\_LIBRARY\_PATH on macOS), which is used 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 one or more directories, separated by colons.

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 [Editing the Active CPF](#)).

# MaxServerConn

Set the maximum number of connections from ECP clients.

## Synopsis

```
[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 instance simultaneously. This is the maximum number of connections that this instance may accept when acting as an ECP server. If you modify this setting, you must restart the instance for the change to take effect.

When an instance is a member of a [sharded cluster](#), this setting must be equal to or greater than the number of nodes in the cluster.

## 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 [Editing the Active CPF](#)).

# MaxServers

---

Set the maximum number of connections to ECP servers.

## Synopsis

```
[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 instance. This is the maximum number of connections that this instance can establish when acting as an ECP client. If you modify this setting, you must restart the instance for the change to take effect.

When an instance is a member of a [sharded cluster](#), this setting must be equal to or greater than the number of nodes in the cluster.

## 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 [Editing the Active CPF](#)).

# Path

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

## Synopsis

```
[config]    Path=directory:directory2:directory3[...]
```

*directory* is the full path to a valid directory. The maximum length for Path is 1024 characters. By default, no directories are listed.

## Description

As part of InterSystems IRIS® data platform startup, UNIX® systems assign new processes a UNIX® PATH environment variable. By default, the UNIX® PATH variable is:

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

InterSystems IRIS processes that use PATH include system daemons, processes started by SYSTEM^%ZSTART, and processes the Super Server creates (such as JDBC/ODBC servers).

Customer applications may require that the UNIX® 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 CPF 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 one or more directories, separated by colons.

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 [Editing the Active CPF](#)).

# PythonPath

Specify directories to be added to `sys.path` for Embedded Python.

## Synopsis

```
[config]    PythonPath=directory
```

*directory* is one or more paths to valid directories. The maximum length for `PythonPath` is 1024 characters. By default, no directories are listed.

## Description

When importing modules, Embedded Python uses the `sys.path` variable to determine the locations where Python packages may be located. `PythonPath` specifies additional directories where you plan to install Python packages, so that Embedded Python can find them when you want to import a module.

You may specify multiple directories. On Windows, separate multiple directories using a semicolon; on other platforms, use a colon to separate multiple directories.

## Example

On Ubuntu, `PythonPath=/libraries/mypackages:/libraries/morepackages` adds two directories to `sys.path`.

The example below, from Terminal, shows that the specified directories are present in `sys.path` when Embedded Python is launched.

```
USER>do ##class(%SYS.Python).Shell()

Python 3.10.13 (main, Aug 25 2023, 13:20:03) [GCC 9.4.0] on linux
Type quit() or Ctrl-D to exit this shell.
>>> import sys
>>> sys.path
['/usr/lib/python3.10.zip', '/usr/lib/python3.10', '/usr/lib/python3.10/lib-dynload',
'/InterSystems/IRIS/lib/python', '/InterSystems/IRIS/mgr/python', '/libraries/mypackages',
'/libraries/morepackages', '/bin/local/lib/python3.10/dist-packages', '/bin/lib/python3/dist-packages',
'/bin/lib/python3.10/dist-packages']
```

## Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration** > **Configuration** > **Additional Settings** > **Advanced Memory**), in the **PythonPath** row, select **Edit**. Enter one or more directories.

Instead of using the Management Portal, you can change `PythonPath` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

When you edit this setting, it takes effect immediately. The new path is used for any new processes.

# PythonRuntimeLibrary

Specify the location of the runtime library to use when running Embedded Python.

## Synopsis

```
[config]    PythonRuntimeLibrary=library
```

*library* is the absolute path and filename of a Python runtime library. The maximum length for `PythonRuntimeLibrary` is 1024 characters. By default, no library is listed.

## Description

`PythonRuntimeLibrary` specifies the location of the Python runtime library to use when running Embedded Python.

This location will vary based on your operating system, Python version, and other factors.

Windows example: `PythonRuntimeLibrary=C:\Program Files\Python311\python3.dll` (Python 3.11 on Windows)

Linux example: `PythonRuntimeLibrary=/usr/lib/x86_64-linux-gnu/libpython3.11.so.1.0` (Python 3.11 on Ubuntu 22.04 on the x86 architecture)

This parameter is required in order to use Embedded Python on InterSystems IRIS 2024.2 and higher on Windows. This parameter is needed on other platforms only when overriding the default version of Python.

**Note:** See [Flexible Python Runtime Feature](#) for more information on configuring the version of Python to be used for Embedded Python.

See Other Supported Features for information on which platforms Flexible Python Runtime is available.

## Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration** > **Configuration** > **Additional Settings** > **Advanced Memory**), in the **PythonRuntimeLibrary** row, select **Edit**. Enter the location of the Python runtime library you want to use.

Instead of using the Management Portal, you can change `PythonRuntimeLibrary` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

When you edit this setting, it takes effect immediately. The new runtime library is used for any new processes.

# PythonRuntimeLibraryVersion

---

Specify the version number of the runtime library to use when running Embedded Python.

## Synopsis

```
[config]    PythonRuntimeLibraryVersion=version
```

*version* is the version number of the Python runtime library to be used for Embedded Python. By default, no version is listed.

## Description

`PythonRuntimeLibraryVersion` specifies the version number of the Python runtime library to use when running Embedded Python.

For example: `PythonRuntimeLibraryVersion=3.11`

This parameter is required in order to use Embedded Python on InterSystems IRIS 2024.2 and higher on Windows. This parameter is recommended on all other platforms.

**Note:** See [Flexible Python Runtime Feature](#) for more information on configuring the version of Python to be used for Embedded Python.

See Other Supported Features for information on which platforms Flexible Python Runtime is available.

## Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration** > **Configuration** > **Additional Settings** > **Advanced Memory**), in the `PythonRuntimeLibraryVersion` row, select **Edit**. Enter the version number of the Python runtime library you want to use.

Instead of using the Management Portal, you can change `PythonRuntimeLibraryVersion` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

When you edit this setting, it takes effect immediately. The new runtime library is used for any new processes.



---

# UUIDv1RandomMac

---

Controls how MAC addresses are treated within the UUID infrastructure.

## Synopsis

```
[config]  
UUIDv1RandomMac=1
```

Allowed values are 0 and 1.

## Description

This parameter controls how MAC addresses are treated when generating GUIDs. In particular, if this parameter is 1, that causes a MAC address to be represented by 47 bits of randomness and one high multicast bit as per [RFC4122](#), Section 4.5. InterSystems suggests that you use `UUIDv1RandomMac=1` if you are generating GUIDs where there are multiple instances on one host.

For details, see the `CreateGuid()` method of `%SYSTEM.Util`.

## Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration** > **Configuration** > **Additional Settings** > **Advanced Memory**), in the **UUIDv1RandomMac** row, select **Edit**. Enter 0 or 1 and click **Save**.

Instead of using the Management Portal, you can change `UUIDv1RandomMac` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

When you edit this setting, it takes effect immediately.

# bbsiz

---

Set the maximum memory per process.

## Synopsis

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

*n* is an integer in the range 256—2,147,483,647 (KB). The default value is -1, which sets the parameter to its maximum value (2,147,483,647 KB).

## 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.

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 [Editing the Active CPF](#)).

When you edit this setting, the updated value applies for all new processes.

When set to the default of -1, or the maximum of about 2 terabytes, maximum size of per process memory is essentially unlimited, as it is very unlikely that a process would approach the use of that much memory.

---

# console

---

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

## Synopsis

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

*ConsoleFile* is the full path to the messages.log file. The maximum length for *ConsoleFile* is 227 characters. By default, no directory is listed.

## Description

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

### *ConsoleFile*

The path to the messages.log file, where InterSystems IRIS® data platform logs messages. Change `ConsoleFile` to specify a new location for the log file; however, the name of the log file must always be `messages.log`. If no value is specified, InterSystems IRIS writes to *install-dir/mgr/messages.log*.

`ConsoleFile` is the second comma-separated 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.

### *VMSConsoleTerminal*

Not in use.

## 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 [Editing the Active CPF](#)).

## errlog

---

Set the maximum number of entries in the error log.

### Synopsis

```
[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](#) 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 [Editing the Active CPF](#)).

# globals

Allocate shared memory to the database cache.

## Synopsis

```
[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, as they are by default, the system automatically allocates 25% of total physical memory. On a 64-bit system, there is a limit of 16 TB.

## 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.

For more information about allocating memory to the database cache, including using the Management Portal to do so, see [Allocating Memory to the Database and Routine Caches](#).

You can also change `globals` using the `Config.config` class or by [editing the CPF](#).

## See Also

- The [routines](#) parameter
- The [DBSizesAllowed](#) parameter in the `[Startup]` section of this reference
- [Shared Memory Allocations](#).
- [Allocating Memory to the Routine and Database Caches](#)
- [Large Block Size Considerations](#)
- [ECP Control Structures](#)

# gmheap

Set the size of the shared memory heap.

## Synopsis

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

*n* is an integer in the range 16384—1,073,741,760 (KB). The default is 0 which will set the size of the shared memory heap to a reasonable value based on the size of the system.

## Description

gmheap is the size in kilobytes of the shared memory heap (formerly known as the generic memory heap) 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 [Shared Memory Heap Usage](#) for more information. Although this page displays memory allocation and use in bytes, bear in mind that shared memory is allocated in pages.

By default gmheap is set to 0. This allows the system to select a reasonable value based on the overall system size. If 0 is selected, the system will configure the size of gmheap to be 3% of the total memory configured for [global buffers](#). The minimum gmheap will be configured to is 307,200 KB (300 MB) and the maximum is 2,097,000 KB (2 GB), however you can still manually configure a larger or smaller amount of memory.

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 shared memory heap size; see [System Requirements for Parallel Dejournaling](#) for more information.

- When parallel SQL query execution is in use

Parallel query execution uses additional shared memory from the shared memory heap, and an increase in gmheap may therefore be required to optimize parallel query performance. See [Shared Memory Considerations](#) 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 [Editing the Active CPF](#)).

If you edit the gmheap setting, you must restart InterSystems IRIS to apply the change. You may also need to increase the allocated huge or large pages when increasing gmheap.

---

# history

---

Define command line recall options.

## Synopsis

```
[config]    history=n
```

*n* is an integer in the range 0—1000. The default value is 500.

## Description

`history` is the maximum number of entries held in the command line/read line recall buffer.

## Changing This Parameter

On the **Advanced Memory** page of the Management Portal (**System Administration** > **Configuration** > **Additional Settings** > **Advanced Memory**), edit the values for **history**.

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 [Editing the Active CPF](#)).

This setting does not require a restart to take effect.

## ijcbuf

---

Set the size of an InterJob Communication buffer.

### Synopsis

```
[config]    ijcbuf=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 [Interprocess Communication](#). 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 [Editing the Active CPF](#)).

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



# ijcnum

---

Set the number of InterJob Communication devices.

## Synopsis

```
[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 `ijcbuf`. For details, see [Interprocess Communication](#).

## 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 [Editing the Active CPF](#)).

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

# jrnbufs

---

Allocate memory for journal buffers.

## Synopsis

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

*n* is an integer. The maximum value is 1024 (MB), and the minimum is 16 for Unicode instances and 8 for 8-bit instances. 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.

Increasing `jrnbufs` also potentially means reducing the journal file size (as configured in the [Management Portal](#), using `^JRNOPTS`, or by setting the `FileSizeLimit` parameter) due a combined total size limit of 4 GB for the two settings; see [Configuring Journal Settings](#) for more information.

## 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 [Editing the Active CPF](#)).

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.

## Synopsis

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

*n* is an integer with a minimum value of 65,536 (bytes). The default value is 0 which is designed to be appropriate for all systems. If `locksiz` is set to 0, it will be limited only by the size shared memory given by the [gmheap](#) parameter.

## Description

By default, `locksiz` is configured to an amount that is designed to be appropriate for all systems. However, since memory used to allocate locks is taken from [gmheap](#) (the Shared 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. Alternatively, you can lower the [LockThreshold](#) to use less space in the lock table.

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 [Editing the Active CPF](#)).

## See Also

- [Locking and Concurrency Control](#)
- The [gmheap](#) parameter
- The [LockThreshold](#) parameter

# memlock

Specify locking shared memory or the text segment into memory.

## Synopsis

```
[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 allocates a shared memory segment for use by control structures, global buffers, routine buffers, and shared 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 [Editing the Active CPF](#)).

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

## netjob

---

Allow remote job requests.

### Synopsis

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

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

### Description

When `netjob` 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 `netjob` in the `Config.config` class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

---

# nlstab

---

Set the number of NLS collation tables.

## Synopsis

```
[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 [Editing the Active CPF](#)).

## overview

---

Review instance platform and version summary.

### Synopsis

```
[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® data platform 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 to the routines cache

## Synopsis

```
[config] routines=n ; This is the format for single-value automatic allocation.
[config] routines=n1, n2, n3, n4, n5, n6 ; This is the format for multiple-value manual allocation.
```

The default for this parameter is `routines=0`. This allocates memory equal to 10% of the memory allocated to 8-kilobyte global buffers in the database cache by the [globals](#) parameter, with a minimum of 80 MB and a maximum of 1 GB.

## Description

The `routines` parameter specifies the amount of memory to allocate for buffering routines. The total memory is shared across six *pools* of different buffer sizes, from 2-kilobyte to 64-kilobyte,

You can configure routine buffers pools in one of two ways — either specify a [single value](#) for automatic allocation, or [multiple values](#) to allocate buffers manually. For typical production instances, automatic allocation is sufficient; however, the ideal allocation for a given application depends on many factors, and adjustment may be necessary to optimize performance.

### Automatic (Single-value) Buffer Allocation

You can specify a single value, *n*, as the total number of megabytes that InterSystems IRIS® data platform allocates for routine buffers. The minimum size is 80 MB; if a smaller value is specified, the instance adjusts up to 80 MB. A value of 0 (the default) allocates memory equal to 10% of the memory allocated to 8-kilobyte global buffers in the database cache by the [globals](#) parameter, with a minimum of 80 MB and a maximum of 1020 MB.

InterSystems IRIS divides the memory among the 4, 16, and 64-kilobyte buffer pools as follows:

- 12.5% in 4-kilobyte buffers
- 37.5% in 16-kilobyte buffers
- 50% in 64-kilobyte buffers

For example, if you specify `routines=500`, InterSystems IRIS creates:

- 16,000 4-kilobyte buffers (62.5 MB)
- 12,000 16-kilobyte buffers (187.5 MB)
- 4,000 64-kilobyte buffers (250 MB)

### Manual (Multi-value) Buffer Allocation

You can specify the amount of memory to allocate for each routine buffer pool. If you do, you must specify all six values; if fewer than six are given, InterSystems IRIS reverts to the automatic format described above, using the first value provided. The six values represent the following:

- *n1* is the number of MB allocated for 2-kilobyte routine buffers.
- *n2* is the number of MB allocated for 4-kilobyte routine buffers.
- *n3* is the number of MB allocated for 8-kilobyte routine buffers.
- *n4* is the number of MB allocated for 16-kilobyte routine buffers.
- *n5* is the number of MB allocated for 32-kilobyte routine buffers.

- *n6* is the number of MB allocated for 64-kilobyte routine buffers. The instance always allocates at least 430 64 KB routine buffers, regardless of the value of *n6*.

For example, if you specify `routines=0,128,128,0,0,800`, InterSystems IRIS creates:

- 128 MB of 4-kilobyte buffers
- 128 MB of 8-kilobyte buffers
- 800 MB of 64-kilobyte buffers

While it is possible to allocate 0 buffers of a particular size (except for 64 KB), the next minimum value is 430; if a smaller number is specified, the instance allocates 430. The combined maximum number of buffers is 33,554,432. The format for InterSystems IRIS routines does not allow more than 32,768 characters for literal strings regardless of the setting for the maximum routine size.

## Changing This Parameter

For more information about allocating memory to the routine cache, including using the Management Portal to do so, see [Allocating Memory to the Database and Routine Caches](#).

You can also change `routines` using the `Config.config` class or by [editing the CPF](#).

## See Also

- The [globals](#) parameter.
- [Shared Memory Allocations](#)
- [Allocating Memory to the Routine and Database Caches](#)

# udevtabsiz

---

Set the maximum size of the device table.

## Synopsis

```
[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 [Editing the Active CPF](#)).

---

# wjdir

---

Set the Write Image Journal files directory.

## Synopsis

```
[config]    wjdir=n
```

*n* is the full path to a valid directory. The maximum length is 226 characters. By default, no directory is listed.

## Description

wjdir is the name of the directory where the write image journal file is stored. 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 wjdir in the Config.config class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

# targwjsz

---

Set the desired size of the wij.

## Synopsis

```
[config]    targwjsz=n
```

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

## Description

targwjsz allocates a number of MB for the Write Image Journal (WIJ) file. The value is an integer, but fractional input is accepted and silently truncated (for example, 35.5 becomes 35).

When targwjsz is set to 0, the WIJ grows as needed, based on activity. The targwjsz parameter allows you to allocate space to the WIJ ahead of time so it does not need to grow during a period of high activity. While there is no upper limit for targwjsz, it is not useful to set it higher than the maximum size of the [database cache](#), as that is the most space the WIJ needs.

For more information, see [Write Image Journaling and Recovery](#).

**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 instance may encounter problems. Allocating space for WIJ is an advanced configuration setting. If you encounter issues with this, contact the InterSystems [Worldwide Response 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 **targwjsz** row. The **Edit: targwjsz** page provides details about the setting and allows you to change its value.

Instead of using the Management Portal, you can change targwjsz in the Config.config class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

# zfheap

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

## Synopsis

```
[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 (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 [Creating a Callout Library](#).

### 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 1-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:

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

### 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 [Editing the Active CPF](#)).

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.

### Synopsis

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

*n* is a string of numbers up to 7 characters long, usually in dot format.

### Description

`Version` is the configuration parameter file (CPF) 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, use [Comments](#) as described in “Introduction to the Configuration Parameter File”.

# [Databases]

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

# Database

Define InterSystems IRIS® data platform databases.

## Synopsis

```
[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 instance name (empty for local instance). If the instance is remote (nonempty name), subsequent fields are ignored. Default is the local instance.
- *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 IRISYS, IRISLIB, IRISTEMP, and IRISAUDIT databases must be local to the instance and mounted 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]
IRISYS=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\,SALESERVER
; Database is on instance SALESERVER
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:PRDDCIFC,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.

## Synopsis

```
[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 *irisfpid.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 **GRETRELEASE**) 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 *irisfpid.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 *irismpid.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 [Editing the Active CPF](#)). If the CPF does not already contain the [Debug] section, you must add it yourself. For example:

```
[ConfigFile]
...

[Debug]      // Add the debug section if necessary
dumpstyle=2  // then, specify a value for dumpstyle.

[Databases]
...
```

# Semsperset

---

Set number of semaphores allocated per set.

## Synopsis

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

*n* is an integer. There is no maximum value. The default value is 0.

## Description

Semsperset is the number of semaphores InterSystems IRIS® data platform should allocate per *semaphore set*. When this parameter is set to 0 (the default), InterSystems IRIS allocates the maximum number of semaphores to each set, which results in the minimum number of semaphore sets.

A larger value for semsperset results in fewer semaphore sets of a larger size. There may be a performance benefit to having fewer semaphores per set, particularly on Linux systems (for example, by setting semsperset equal to 250), although this has not been thoroughly tested.

For more information, see [Semaphores in InterSystems Products](#).

## Changing This Parameter

You can change semsperset by editing the CPF in a text editor (as described in [Editing the Active CPF](#)). If the CPF does not already contain the [Debug] section, you must add it yourself. For example:

```
[ConfigFile]
...

[Debug]      // Add the debug section if necessary,
semsperset=250 // then specify a value for semsperset.

[Databases]
...
```

# [Devices]

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

# Devices

Define device types.

## Synopsis

[Devices]      *Name=a^b^c^d^e^f^g^h*

## Description

The [Devices] section of the configuration parameter file (CPF) 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 [Allowing Users to Specify a Device](#), especially the section [%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.

## Synopsis

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

## Description

The [DeviceSubTypes] section of the configuration parameter file (CPF) 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 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 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 $X=DX,$Y=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)_(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-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.

## Synopsis

```
[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 [Editing the Active CPF](#)).

---

# ClientReconnectInterval

---

Set interval between ECP reconnection attempts.

## Synopsis

```
[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 [Editing the Active CPF](#)).

# ServerTroubleDuration

---

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

## Synopsis

```
[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 [Editing the Active CPF](#)).

# [ECPServers]

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

# ECPServers

Define ECP servers.

## Synopsis

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

## 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](#) 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.
- *SSLConfig* – If 1, connections to the server use TLS/SSL.
- *BatchMode* – If 1, the server process for this data server runs in batch mode. In batch mode, the data server always loads blocks and caches them in batch level.

## 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 using the Config.ECPServers class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).



# [Gateways]

This topic describes the [Gateways] section of the CPF, which contains entries for each gateway configured on the InterSystems IRIS® data platform instance and its host system. Gateways, which are also known as external language servers, provide fully integrated bi-directional connections between InterSystems IRIS and external language platforms.

Each definition consists of a name followed by = and then a comma-separated list of values defining the gateway; some fields are required, and others can be left blank, in which case default values are used. As shown in the following, there are seven types of gateways defined in the default CPF of a newly installed instance: .NET, IntegratedML, JDBC, Java, Python, R, and XSLT. In addition to these local gateway types, you can define a remote gateway on a remote system.

```
[Gateways]
%DotNet Server=.NET,53398,%Gateway_Object,N6.0
%IntegratedML Server=ML,53598,%Gateway_ML
%JDBC Server=JDBC,53798,%Gateway_SQL
%Java Server=Java,53298,%Gateway_Object
%Python Server=Python,53498,%Gateway_Object
%R Server=R,53898,%Gateway_Object
%XSLT Server=XSLT,53698,%Gateway_Object
```

**Note:** JDBC, Java, IntegratedML, XSLT, and R gateways are all Java-based and use the same fields in their definitions.

The *Type* and *Port* fields are required for all gateway types. The third field, *Resource*, specifies the [gateway resource](#) a user must have USE permission on to use the gateway; for example, the default %JDBC Server definition above specifies that a user must have the %Gateway\_SQL:USE privilege to use the JDBC gateway. InterSystems strongly recommends protecting all gateways by specifying an appropriate resource.

If you have the [%Admin\\_ExternalLanguageServerEdit:USE privilege](#), which by default is held by the %Manager role, you can use the following methods to create a new gateway of any type or to edit or delete an existing gateway:

- The **External Language Servers** page of the Management Portal (**System Administration > Configuration > Connectivity > External Language Servers**).
- The Config.Gateway API.
- Modifying the CPF of the instance using the [iris merge](#) or by [editing it directly](#).

You can start or stop a gateway and display its activity log using the **External Language Servers** page or the \$system.external interface.

Once a gateway has been defined, either in the default CPF or by a user, its name cannot be changed.

# %DotNet Server

Define a .NET gateway.

## Synopsis

```
[Gateways]
%DotNet Server=.NET,Port,[optional fields]
```

## Description

%DotNet Server is the name of the .NET gateway definition in the default CPF, but additional .NET gateways with different names can be created and existing gateways can be edited or deleted as described in [\[Gateways\]](#). Each definition consists of a name followed by = and then a comma-separated list of values defining the gateway; some fields are required, and others can be left blank, in which case default values are used. Fields specific to .NET gateways and those that can be used with all gateways are described in the following two sections. For a more detailed description, see .NET External Server Setup and Defining External Server Configurations for .NET.

## .NET Gateway Fields

The first three fields in a .NET gateway definition are common to all gateways and the next three are specific to .NET gateways. These are shown in order in the following:

- *Type* (required) — Type of the gateway. When defining a .NET gateway, this is set to .NET.
- *Port* (required) — TCP port number for communication between the gateway and the proxy classes in InterSystems IRIS. This port number must not conflict with any other local TCP port on the server.
- *Resource* — The [gateway resource](#) that controls access to this gateway; in the default CPF, this is %Gateway\_Object.

**Important:** If no resource is specified, the gateway is public, but InterSystems strongly recommends protecting all gateways using this mechanism.

- *DotNetVersion* — Specifies the .NET version to be used. Possible values are N6.0, N5.0, F4.6.2, F4.5, F3.5, F2.0, and C2.1; the default is N6.0.

**Note:** The default of N6.0 is specified in the default CPF to ensure it is set when an instance is upgraded from a prior version in which the default was F4.5.

- *FilePath* — Location of the gateway executable, used in assembling the command to start the gateway on the local server. If this setting is not specified, the default directory *install-dir\dev\dotnet\bin\* is used, with the appropriate sub-directory selected according to the *DotNetVersion* setting, for example if *DotNetVersion* is N6.0 the *FilePath* would be *install-dir\dev\dotnet\bin\net6.0*.

**Note:** Executables for *DotNetVersion* values F4.5, F2.0 and C2.1 are not installed with the current version of InterSystems IRIS, but the values are provided to support use of these executables from an earlier version.

- *Exec32* — On 64-bit platforms, a value of 1 indicates that the gateway is to be executed as 32-bit; the default is 0, for 64-bit.

## General Gateway Fields

The remaining eight fields can be used in any gateway definition, and are shown in order in the following:

- *SSLConfigurationServer* — Name of the SSL/TLS configuration to be used for server TLS/SSL.
- *SSLConfigurationClient* — Name of the SSL/TLS configuration to be used for client TLS/SSL.

- *VerifySSLHostName* — Whether the TLS/SSL client should perform hostname verification. The default is 0 (no hostname verification).
- *UseSharedMemory* — Whether to use shared memory for connection if available. The default is 1 (use shared memory).
- *LogFile* — Full pathname of the file used to log all communication between InterSystems IRIS and the gateway. This optional property should only be used when debugging. The maximum length is 1023 characters.
- *AllowedIpAddresses* — IP addresses that allow incoming connections. Specify 0.0.0.0 to allow connections on all IP addresses local to the machine (127.0.0.1, VPN address, and so on). You can also specify a single existing local IP address to restrict connections to that IP address. The default is 127.0.0.1.
- *ConnectionTimeout* — Number of seconds to wait for a connection with the gateway to be established. The range is 2-300; the default value is 5.
- *InitializationTimeout* — Number of seconds to wait for a response during initialization of the gateway. The range is 2-300; the default value is 5.

## Changing This Parameter

For information about changing and adding gateway definitions, see [\[Gateways\]](#).

# %IntegratedML Server

Define an IntegratedML gateway.

## Synopsis

```
[Gateways]
%IntegratedML Server=ML,Port,[optional fields]
```

## Description

%IntegratedML Server is the name of the IntegratedML gateway definition in the default CPF, but additional IntegratedML gateways with different names can be created and existing gateways can be edited or deleted as described in [\[Gateways\]](#). Each definition consists of a name followed by = and then a comma-separated list of values defining the gateway; some fields are required, and others can be left blank, in which case default values are used. Fields specific to .IntegratedML (and other Java-based) gateways and those that can be used with all gateways are described in the following two sections. For a more detailed description, see [Java External Server Setup and Defining External Server Configurations for Java](#).

## IntegratedML Gateway Fields

The first three fields in an IntegratedML gateway definition are common to all gateways and the next three are specific to Java-based gateways. These are shown in order in the following:

- *Type* (required) — Type of the gateway. When defining an IntegratedML gateway, this is set to ML.
- *Port* (required) — TCP port number for communication between the gateway and the proxy classes in InterSystems IRIS. This port number must not conflict with any other local TCP port on the server.
- *Resource* — The [gateway resource](#) that controls access to this gateway; for %Integrated\_ML Server in the default CPF, this is %Gateway\_ML.

**Important:** If no resource is specified, the gateway is public, but InterSystems strongly recommends protecting all gateways using this mechanism.

- *ClassPath* — The files that must be passed as an argument when starting the Java Virtual Machine (JVM), which are typically the files containing the classes used via the gateway. The classpath must be properly quoted, with the correct platform-specific separators between files.
- *JVMArgs* — Optional arguments passed to the JVM, used in the command to start the gateway.
- *JavaHome* — Location of the JVM, used in the command to start the Gateway.

**Important:** This field is required unless a default JVM is set for the host system.

## General Gateway Fields

The remaining eight fields can be used in any gateway definition, and are shown in the [General Gateway Fields section under %DotNet Server](#).

## Changing This Parameter

For information about changing and adding gateway definitions, see [\[Gateways\]](#).

# %JDBC Server

Define a JDBC gateway.

## Synopsis

```
[Gateways]
%JDBC Server=JDBC,Port,[optional fields]
```

## Description

%JDBC Server is the name of the JDBC gateway definition in the default CPF, but additional JDBC gateways with different names can be created and existing gateways can be edited or deleted as described in [\[Gateways\]](#). Each definition consists of a name followed by = and then a comma-separated list of values defining the gateway; some fields are required, and others can be left blank, in which case default values are used. Fields specific to JDBC (and other Java-based) gateways and those that can be used with all gateways are described in the following two sections. For a more detailed description, see [Java External Server Setup and Defining External Server Configurations for Java](#).

## JDBC Gateway Fields

The first three fields in a JDBC gateway definition are common to all gateways and the next three are specific to Java-based gateways. These are shown in order in the following:

- *Type* (required) — Type of the gateway. When defining a JDBC gateway, this is set to JDBC.
- *Port* (required) — TCP port number for communication between the gateway and the proxy classes in InterSystems IRIS. This port number must not conflict with any other local TCP port on the server.
- *Resource* — The [gateway resource](#) that controls access to this gateway; for %JDBC Server in the default CPF, this is %Gateway\_SQL.

**Important:** If no resource is specified, the gateway is public, but InterSystems strongly recommends protecting all gateways using this mechanism.

- *ClassPath* — The files that must be passed as an argument when starting the Java Virtual Machine (JVM), which are typically the files containing the classes used via the gateway. The classpath must be properly quoted, with the correct platform-specific separators between files.
- *JVMArgs* — Optional arguments passed to the JVM, used in the command to start the gateway.
- *JavaHome* — Location of the JVM, used in the command to start the Gateway.

**Important:** This field is required unless a default JVM is set for the host system.

## General Gateway Fields

The remaining eight fields can be used in any gateway definition, and are shown in the [General Gateway Fields section under %DotNet Server](#).

## Changing This Parameter

For information about changing and adding gateway definitions, see [\[Gateways\]](#).

# %Java Server

Define a Java gateway.

## Synopsis

```
[Gateways]
%Java Server=Java,Port,[optional fields]
```

## Description

`%Java Server` is the name of the Java gateway definition in the default CPF, but additional Java gateways with different names can be created and existing gateways can be edited or deleted as described in [\[Gateways\]](#). Each definition consists of a name followed by = and then a comma-separated list of values defining the gateway; some fields are required, and others can be left blank, in which case default values are used. Fields specific to Java (and other Java-based) gateways and those that can be used with all gateways are described in the following two sections. For a more detailed description, see [Java External Server Setup and Defining External Server Configurations for Java](#).

## Java Gateway Fields

The first three fields in a Java gateway definition are common to all gateways and the next three are specific to Java-based gateways. These are shown in order in the following:

- *Type* (required) — Type of the gateway. When defining a Java gateway, this is set to `Java`.
- *Port* (required) — TCP port number for communication between the gateway and the proxy classes in InterSystems IRIS. This port number must not conflict with any other local TCP port on the server.
- *Resource* — The [gateway resource](#) that controls access to this gateway; for `%Java Server` in the default CPF, this is `%Gateway_Object`.

**Important:** If no resource is specified, the gateway is public, but InterSystems strongly recommends protecting all gateways using this mechanism.

- *ClassPath* — The files that must be passed as an argument when starting the Java Virtual Machine (JVM), which are typically the files containing the classes used via the gateway. The classpath must be properly quoted, with the correct platform-specific separators between files.
- *JVMArgs* — Optional arguments passed to the JVM, used in the command to start the gateway.
- *JavaHome* — Location of the JVM, used in the command to start the Gateway.

**Important:** This field is required unless a default JVM is set for the host system.

## General Gateway Fields

The remaining eight fields can be used in any gateway definition, and are shown in the [General Gateway Fields section under %DotNet Server](#).

## Changing This Parameter

For information about changing and adding gateway definitions, see [\[Gateways\]](#).

# %Python Server

Define a Python gateway.

## Synopsis

```
[Gateways]
%Python Server=Python,Port,[optional fields]
```

## Description

`%Python Server` is the name of the Python gateway definition in the default CPF, but additional Python gateways with different names can be created and existing gateways can be edited or deleted as described in [\[Gateways\]](#). Each definition consists of a name followed by = and then a comma-separated list of values defining the gateway; some fields are required, and others can be left blank, in which case default values are used. Fields specific to Python gateways and those that can be used with all gateways are described in the following two sections. For a more detailed description, see [Python External Server Setup and Defining External Server Configurations for Python](#).

## Python Gateway Fields

The first three fields in a Python gateway definition are common to all gateways and the next three are specific to Python gateways. These are shown in order in the following:

- *Type* (required) — Type of the gateway. When defining a Python gateway, this is set to `Python`.
- *Port* (required) — TCP port number for communication between the gateway and the proxy classes in InterSystems IRIS. This port number must not conflict with any other local TCP port on the server.
- *Resource* — The [gateway resource](#) that controls access to this gateway; for `%Python Server` in the default CPF, this is `%Gateway_SQL`.

**Important:** If no resource is specified, the gateway is public, but InterSystems strongly recommends protecting all gateways using this mechanism.

- *PythonPath* — Location of the Python interpreter, used in assembling the command to start the gateway on the local server.

**Important:** This field is required unless a default Python interpreter is set for the host system.

- *PythonOptions* — Python options to be included in the command to start the gateway on the local server.
- (not used) — Leave this field empty (consecutive commas).

## General Gateway Fields

The remaining eight fields can be used in any gateway definition, and are shown in the [General Gateway Fields section under %DotNet Server](#).

## Changing This Parameter

For information about changing and adding gateway definitions, see [\[Gateways\]](#).

## %R Server

Define an R gateway.

### Synopsis

```
[Gateways]  
%R Server=R,Port,[optional fields]
```

### Description

`%R Server` is the name of the R gateway definition in the default CPF, but additional R gateways with different names can be created and existing gateways can be edited or deleted as described in [\[Gateways\]](#). Each definition consists of a name followed by = and then a comma-separated list of values defining the gateway; some fields are required, and others can be left blank, in which case default values are used. Fields specific to R (and other Java-based) gateways and those that can be used with all gateways are described in the following two sections. For a more detailed description, see [Java External Server Setup](#) and [Defining External Server Configurations for Java](#).

### R Gateway Fields

The first three fields in an R gateway definition are common to all gateways and the next three are specific to Java-based gateways. These are shown in order in the following:

- *Type* (required) — Type of the gateway. When defining an R gateway, this is set to R.
- *Port* (required) — TCP port number for communication between the gateway and the proxy classes in InterSystems IRIS. This port number must not conflict with any other local TCP port on the server.
- *Resource* — The [gateway resource](#) that controls access to this gateway; for `%R Server` in the default CPF, this is `%Gateway_Object`.

**Important:** If no resource is specified, the gateway is public, but InterSystems strongly recommends protecting all gateways using this mechanism.

- *ClassPath* — The files that must be passed as an argument when starting the Java Virtual Machine (JVM), which are typically the files containing the classes used via the gateway. The classpath must be properly quoted, with the correct platform-specific separators between files.
- *JVMArgs* — Optional arguments passed to the JVM, used in the command to start the gateway.
- *JavaHome* — Location of the JVM, used in the command to start the Gateway.

**Important:** This field is required unless a default JVM is set for the host system.

### General Gateway Fields

The remaining eight fields can be used in any gateway definition, and are shown in the [General Gateway Fields](#) section under `%DotNet Server`.

### Changing This Parameter

For information about changing and adding gateway definitions, see [\[Gateways\]](#).



# %XSLT Server

Define an XSLT gateway.

## Synopsis

```
[Gateways]
%XSLT Server=XSLT,Port,[optional fields]
```

## Description

%XSLT Server is the name of the XSLT gateway definition in the default CPF, but additional XSLT gateways with different names can be created and existing gateways can be edited or deleted as described in [\[Gateways\]](#). Each definition consists of a name followed by = and then a comma-separated list of values defining the gateway; some fields are required, and others can be left blank, in which case default values are used. Fields specific to XSLT (and other Java-based) gateways and those that can be used with all gateways are described in the following two sections. For a more detailed description, see [Java External Server Setup](#) and [Defining External Server Configurations for Java](#).

## XSLT Gateway Fields

The first three fields in an XSLT gateway definition are common to all gateways and the next three are specific to Java-based gateways. These are shown in order in the following:

- *Type* (required) — Type of the gateway. When defining an XSLT gateway, this is set to XSLT.
- *Port* (required) — TCP port number for communication between the gateway and the proxy classes in InterSystems IRIS. This port number must not conflict with any other local TCP port on the server.
- *Resource* — The [gateway resource](#) that controls access to this gateway; for %XSLT Server in the default CPF, this is %Gateway\_Object.

**Important:** If no resource is specified, the gateway is public, but InterSystems strongly recommends protecting all gateways using this mechanism.

- *ClassPath* — The files that must be passed as an argument when starting the Java Virtual Machine (JVM), which are typically the files containing the classes used via the gateway. The classpath must be properly quoted, with the correct platform-specific separators between files.
- *JVMArgs* — Optional arguments passed to the JVM, used in the command to start the gateway.
- *JavaHome* — Location of the JVM, used in the command to start the Gateway.

**Important:** This field is required unless a default JVM is set for the host system.

## General Gateway Fields

The remaining eight fields can be used in any gateway definition, and are shown in the [General Gateway Fields](#) section under %DotNet Server.

## Changing This Parameter

For information about changing and adding gateway definitions, see [\[Gateways\]](#).

# Remote Server

Define a remote gateway.

## Synopsis

```
[Gateways]
Name=Remote,Port,Resource,Address,(not used),(not used),[optional fields]
```

## Description

There is no remote server defined in the [Gateways] section of the default CPF, but remote gateways can be created and existing gateways can be edited or deleted as described in [\[Gateways\]](#). Each definition consists of a name followed by = and then a comma-separated list of values defining the gateway; some fields are required, and others can be left blank, in which case default values are used. Fields specific to remote gateways and those that can be used with all gateways are described in the following two sections.

## Remote Gateway Fields

The first three fields in a remote gateway definition are common to all gateways and the next three are specific to remote gateways. These are shown in order in the following:

- *Type* (required) — Type of the gateway. When defining a remote gateway, this is set to `remote`.
- *Port* (required) — TCP port number for communication between the gateway and the proxy classes in InterSystems IRIS. This port number must not conflict with any other local TCP port on the remote server.
- *Resource* — The [gateway resource](#) that controls access to this gateway. For a remote gateway created using the Management Portal (see [\[Gateways\]](#)), the default is `%Gateway_Object`.

**Important:** If no resource is specified, the gateway is public, but InterSystems strongly recommends protecting all gateways using this mechanism.

- *Address* — The IP address or hostname of the remote system on which you are defining the gateway.
- (not used) — Leave this field empty (consecutive commas).
- (not used) — Leave this field empty (consecutive commas).

## General Gateway Fields

The remaining eight fields can be used in any gateway definition, and are shown in the [General Gateway Fields section under %DotNet Server](#).

## Changing This Parameter

For information about changing and adding gateway definitions, see [\[Gateways\]](#).

# [IO]

The parameters found in the [IO] section of the CPF allow you to change the default mnemonic (^%X364) for WRITE commands. You can also find these settings on the **IO Settings** page of the Management Portal (**System Administration** > **Configuration** > **Device Settings** > **IO Settings**).

---

## Other

---

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

### Synopsis

```
[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 [Controlling Devices with Mnemonic Spaces](#).

### 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 [Editing the Active CPF](#)).

---

# File

---

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

## Synopsis

```
[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 [Controlling Devices with Mnemonic Spaces](#).

## 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 [Editing the Active CPF](#)).

---

# Terminal

---

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

## Synopsis

```
[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 [Controlling Devices with Mnemonic Spaces](#).

## 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 [Editing the Active CPF](#)).

# [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.

## Synopsis

```
[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 an empty string. After InterSystems IRIS® data platform starts for the first time, the actual journal directory is filled in here, such as *<install-dir>\mgr\journal*.

The combined length of AlternateDirectory and the [JournalFilePrefix](#) parameters must be less than 208 characters. (The maximum length of JournalFilePrefix is 64 characters.)

## 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), from the ^JOURNAL utility (as described in [Update Journal Settings Using ^JRNOPTS](#)), or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).



---

# ArchiveName

---

Specify where to copy journal files when archiving them.

## Synopsis

```
[Journal]      ArchiveName=archivename
```

Where *archivename* is the name of an [archive target](#). There is no default.

## Description

ArchiveName is the name of an [archive target](#) defined in the [Archives] section. .

## Changing This Parameter

On the **Journal Settings** page of the Management Portal (**System Administration > Configuration > System Configuration > Journal Settings**), in the **Archive journal files** section, select an existing target from the **To Archive Target** dropdown list.

Instead of using the Management Portal, you can change ArchiveName in the Config.Journal class (as described in the class reference), from the ^JOURNAL utility (as described in [Update Journal Settings Using ^JRNOPTS](#)), or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

# BackupsBeforePurge

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

## Synopsis

[Journal]      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 instance 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.

**Note:** If [PurgeArchived](#) is 1, this setting is ignored.

For details about journal files, see [Overview of Journaling](#).

## 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), from the ^JOURNAL utility (as described in [Update Journal Settings Using ^JRNOPTS](#)), or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

# CompressFiles

Specify whether to compress completed journal files.

## Synopsis

```
[Journal]    CompressFiles=n
```

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

## Description

When `CompressFiles` is enabled ( $n=1$ ), InterSystems IRIS® data platform automatically compresses finished journal files. The system periodically scans for completed journal files and compresses them using Zstd compression. The active journal file is not compressed until it is completed.

Compressed journal files retain the same name, but with a z added to the end. For example, the journal file 20210818.001, when compressed, becomes 20210818.001z. Most InterSystems IRIS functions and utilities are able to access a compressed journal file by its original name, ignoring the z.

For details about journal files, see [Overview of Journaling](#).

## Changing This Parameter

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

Instead of using the Management Portal, you can change `CompressFiles` in the `Config.Journal` class (as described in the class reference), from the **^JOURNAL** utility (as described in [Update Journal Settings Using ^JRNOPTS](#)), or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

# CurrentDirectory

---

Set the primary location of the journal file.

## Synopsis

```
[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 an empty string. After InterSystems IRIS® data platform starts for the first time, the actual journal directory is filled in here, such as *<install-dir>\mgr\journal\*.

The combined length of CurrentDirectory and the [JournalFilePrefix](#) parameters must be less than 208 characters. (The maximum length of JournalFilePrefix is 64 characters.)

## 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), from the ^JOURNAL utility (as described in [Update Journal Settings Using ^JRNOPTS](#)), or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

# DaysBeforePurge

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

## Synopsis

[Journal]      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.

**Note:** If [PurgeArchived](#) is 1, this setting is ignored.

For details about journal files, see [Overview of Journaling](#).

## 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), from the ^JOURNAL utility (as described in [Update Journal Settings Using ^JRNOPS](#)), or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

# FileSizeLimit

---

Set the maximum size of a journal file.

## Synopsis

```
[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 [Journal File Names and Rollover](#).

The FileSizeLimit and [jrnbufs](#) parameters have a maximum combined size limit of 4 GB; see [Configuring Journal Settings](#) for more information.

## 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), from the **^JOURNAL** utility (as described in [Update Journal Settings Using ^JRNOPTS](#)), or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

# FreezeOnError

Allow journaling freeze when a journal I/O error occurs.

## Synopsis

```
[Journal] FreezeOnError=n
```

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

## Description

When `FreezeOnError=0` (false, the default), InterSystems IRIS does not freeze journaling on a journal file I/O error. This option keeps the instance available, but exposes it to potential data loss.

The journal daemon retries the failed operation periodically (typically at one second intervals) until either it succeeds, or journaling is disabled because the instance cannot buffer any further journaled updates or a predetermined time limit (typically 150 seconds) has been reached.

**Important:** When journaling is disabled, you should back up your databases as soon as possible. Continuing without journaling is a calculated risk, as it means the activity that occurs during this period cannot be restored.

Once journaling has been disabled, you must manually restart it, which you can do by running the [^JRNSTART](#) routine or selecting option 1, *Begin Journaling*, from the [^JOURNAL](#) routine menu.

When `FreezeOnError=1` (true), InterSystems IRIS immediately freezes all journaled global updates when a journal file I/O error occurs. Global updates are also frozen if the journal daemon has been unable to complete a journal write for at least 30 seconds. This option protects the instance against data loss, but makes it less available or unavailable while the problem is being resolved.

The journal daemon retries the failed I/O operation and unfreezes global updates after it succeeds. Meanwhile, the freezing of global updates causes other jobs to hang. The typical outcome is that InterSystems IRIS hangs until you resolve the journaling problem, with the system appearing to be down to operational end-users. While InterSystems IRIS is hung you can take corrective measures, such as freeing up disk space, switching the journal to a different disk, or correcting a hardware failure.

**Important:** The `FreezeOnError` setting is automatically overridden and turned on when an instance is a [failover member of a mirror](#).

For further details, see [Journal I/O Errors](#).

## 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 [Editing the Active CPF](#)).

# JournalFilePrefix

---

Set a journal file name prefix.

## Synopsis

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

*n* is an alphanumeric string. The maximum length is 64 characters. 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 Oct20221001.001.

## 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), from the ^JOURNAL utility (as described in [Update Journal Settings Using ^JRNOPTS](#)), or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).



# JournalcspSession

---

Allow journaling of web sessions.

## Synopsis

```
[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 [Editing the Active CPF](#)).

# PurgeArchive

---

Specify whether to purge journal files as soon as they are copied to the archive target.

## Synopsis

[Journal]      PurgeArchive=*boolean*

Where *boolean* is 1 or 0. The default is 0.

## Description

PurgeArchive specifies whether to purge journal files as soon as they are copied to the archive target. Necessary files are never purged, no matter what this setting is.

If PurgeArchive is 1, the settings [BackupsBeforePurge](#) and [DaysBeforePurge](#) are both ignored.

## Changing This Parameter

On the **Journal Settings** page of the Management Portal (**System Administration** > **Configuration** > **System Configuration** > **Journal Settings**), in the **Archive journal files** section, select the **As soon as they are copied to archive** check box. (It is necessary to choose an archive target, in order to see this option.)

Instead of using the Management Portal, you can change PurgeArchive in the Config.Journal class (as described in the class reference), from the ^JOURNAL utility (as described in [Update Journal Settings Using ^JRNOPTS](#)), or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

# [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.

## Synopsis

```
[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 [Configuring InterSystems IRIS Licensing](#).

## 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.

# [Logging]

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

# ChildProcessLaunchCommand

Set the pipe command specifying where the system sends the structured log data.

## Synopsis

[Logging] ChildProcessLaunchCommand=*Executable Options*

The default is of the form:

```
ChildProcessLaunchCommand=irislogd.exe -f /path/to/myfilename.log
```

## Description

The `ChildProcessLaunchCommand` specifies where the system sends structured log data. It takes a target executable that receives the log data. You can optionally specify arguments for the executable.

For the default `irislogd` executable, you can pass the following arguments:

Argument	Purpose
-d	Emit diagnostic and error messages
-e <i>errfilename</i>	Write errors and diagnostic messages to the given file.
-f <i>logfilename</i>	Write log messages to the given file.
-h <i>hostname</i>	Includes the given host name in the structured log file.
-i <i>irisinstance</i>	Includes the given instance name in the structured log file.
-s	Write log messages to the Unix® syslog facility (Unix® only)

## Changing This Parameter

You can change this parameter using the management portal, the **^LOGDMN** routine, or the class-based API (all described in [Configure Structured Logging](#). Alternatively, you can edit the CPF in a text editor (as described in [Editing the Active CPF](#)).

---

# Enabled

---

Allow structured logging data collection.

## Synopsis

```
[Logging] Enabled=n
```

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

## Description

When you enable structured logging (Enabled=1), the log daemon automatically activates when InterSystems IRIS® data platform starts up. However, setting this to 1 will NOT start the log daemon. If this parameter is set to 0, any attempt to start the log daemon fails.

## Changing This Parameter

You can change this parameter using the management portal, the ^LOGDMN routine, or the class-based API (all described in [Configure Structured Logging](#)). Alternatively, you can edit the CPF in a text editor (as described in [Editing the Active CPF](#)).

## Format

---

The format of the data sent to the pipe.

### Synopsis

```
[Logging] Format=n
```

*n* is either name/value pair (NVP) or JSON. The default value is NVP.

### Description

The log daemon can format output for structured logging in NVP or JSON format. In the NVP format, each line in the file contains a set of name/value pairs separated by spaces. Each name/value pair has the form *name=value*, and if value includes a space character, then value is enclosed in parentheses. In the JSON format, each line in the file is a JSON object with a set of properties.

### Changing This Parameter

You can change this parameter using the management portal, the **^LOGDMN** routine, or the class-based API (all described in [Configure Structured Logging](#)). Alternatively, you can edit the CPF in a text editor (as described in [Editing the Active CPF](#)).



---

# Interval

---

Set the interval in seconds between successive calls to the pipe command.

## Synopsis

```
[Logging] Interval=n
```

*n* is a number in the range of 0.01–3600. The default value is 10.

## Description

`Interval` represents the number of seconds the log daemon waits before scanning the `messages.log` file and `%SYS.Audit` log for new entries. It may be expressed as a decimal up to two digit points and can be no less than 0.01 (10 ms) and no larger than 3600 (equivalent to 1 hour).

## Changing This Parameter

You can change this parameter using the management portal, the `^LOGDMN` routine, or the class-based API (all described in [Configure Structured Logging](#)). Alternatively, you can edit the CPF in a text editor (as described in [Editing the Active CPF](#)).

## Level

---

Set the minimum log level.

### Synopsis

```
[Logging] Level=n
```

*n* is one of the following:

- DEBUG2
- DEBUG
- INFO
- WARN
- SEVERE
- FATAL

The default value is WARN.

### Description

This parameter represents the minimum log level of detail, according to the following:

- DEBUG2 — detailed debug messages (such as hex dumps).
- DEBUG — less detailed debug messages.
- INFO — informational messages, including all audit events.
- WARN (the default) — warnings, which indicate problems that may need attention but that have not disrupted operations.
- SEVERE — severe errors, which indicate problems that have disrupted operations.
- FATAL — fatal errors, which indicate problems have caused the system not to run.

### Changing This Parameter

You can change this parameter using the management portal, the **^LOGDMN** routine, or the class-based API (all described in [Configure Structured Logging](#)). Alternatively, you can edit the CPF in a text editor (as described in [Editing the Active CPF](#)).

# [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.

## Synopsis

```
[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 instance 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 [Add Global, Routine, and Package Mapping to a Namespace](#).

## 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.

## Synopsis

```
[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 [Add Global, Routine, and Package Mapping to a Namespace](#).

## 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.

## Synopsis

```
[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 [Add Global, Routine, and Package Mapping to a Namespace](#).

## 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.

## Synopsis

```
[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 instances. 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 instance 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

## Synopsis

```
[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 [Async Mirror Members](#).

## 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 [Editing the Active CPF](#)).

# AsyncMemberType

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

## Synopsis

```
[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 [Async Mirror Members](#).

## 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 [Editing the Active CPF](#)).



# AsyncUseSystemPurgeInterval

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

## Synopsis

```
[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 de journaled.
- 1 - Mirror journal files are purged according to the instances journal file purge criteria.

For more information, see [Editing or Removing an Async Members](#).

## 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 [Editing the Active CPF](#)).

# JoinMirror

---

Specify whether the instance processes its mirror configuration at startup.

## Synopsis

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

*n* is either 1 or 0.

## Description

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

When this parameter is not enabled, the mirror configuration is ignored and the instance is not initialized as a mirror member. This is recommended when there is a problem in the configuration which prevents the instance from starting, or if the member must be reconfigured before joining the mirror. For example, if an instance 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 [Editing the Active CPF](#)).

---

# SystemName

---

Set the name of the mirror member.

## Synopsis

```
[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 instance. 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 [Editing the Active CPF](#)).

## 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.

### Synopsis

```
[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 [Editing the Active CPF](#)).

---

# VirtualAddressInterface

---

When a virtual IP address is configured, this is the name of the local network interface hosting the IP address.

## Synopsis

```
[MirrorMember]    VirtualAddressInterface=InterfaceName
```

## Description

A mirror virtual IP address (VIP) allows all external clients (language bindings, ODBC/JDBC/SQL clients, and so on) to connect to the mirror through a single address. For more information, see the [\[Mirrors\]](#) section and [Configuring a Mirror Virtual IP \(VIP\)](#).

## 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**. Select the **Network Interface** you wish to use.

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 [Editing the Active CPF](#)).



# [Mirrors]

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

# Mirrors

Define mirrors that include this instance.

## Synopsis

```
[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](#) 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 async) 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 [Create and Edit TLS Configurations for a Mirror](#). 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, both failover members must be on the same subnet. For more information, see [Configuring a Mirror Virtual IP \(VIP\)](#).
- *i* (0) – For internal use.
- *j* (ArbiterNode) – The network address of the arbiter configured for this mirror. The arbiter is an independent instance 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 which 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 *CompressionForFailoverMembers*. Possible values are 0 (ZLIB), 1 (ZSTD), and 2 (LZ4).
- *p* (CompressionTypeForAsyncMembers) – Determines the compression type for *CompressionForAsyncMembers*. Possible values are 0 (ZLIB), 1 (ZSTD), and 2 (LZ4).

For more information on Mirroring, see [Mirroring](#).

## 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]

**Important:** The [Miscellaneous] parameters have been retained for compatibility, and all except for `ShutDownLogErrors` should not be used when building new applications.

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

# AsyncDisconnectErr

Allow processes to receive disconnect errors asynchronously.

## Synopsis

[Miscellaneous]      AsyncDisconnectErr=*n*

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

## Description

**Important:**      This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

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

# AsynchError

---

Allow processes to receive asynchronous errors.

## Synopsis

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

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

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

To change this parameter for a single process only (as opposed to instance-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.

## Synopsis

[Miscellaneous]      BreakMode=*n*

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

## Description

**Important:**      This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

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

# CollectResourceStats

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

## Synopsis

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

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

## Description

**Important:** This parameter has been retained for compatibility, but has no effect. Resource statistics are always collected by all instances on all supported platforms.

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

For more information on instance resource statistics, see [Monitoring Performance Using ^mgstat](#) and the `Enumresource` function in [Monitoring InterSystems IRIS Using Web Services](#).

## 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](#) with a text editor.

# DisconnectErr

---

Specify how processes respond to a disconnect.

## Synopsis

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

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

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

`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](#) with a text editor.

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



# FileMode

Allow writing to a non-existent file.

## Synopsis

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

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

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

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

# GlobalKillEnabled

---

Allow KILL of an unsubscribed global.

## Synopsis

[Miscellaneous]      GlobalKillEnabled=*n*

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

## Description

**Important:**      This parameter has been retained for compatibility, but should not be used when building new applications.

Deprecated. When GlobalKillEnabled is enabled ( $n = 1$ ), a KILL of an unsubscribed global is allowed, so you can kill all subscripsts 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](#) with a text editor.

To change this parameter for a single process only (as opposed to instance-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 instance-wide.

## Synopsis

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

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

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

IEEEError sets the \$DOUBLE function return-value behavior instance-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](#) with a text editor.

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

# LicenseAltHeaders

---

Use an alternative set of HTTP headers for client addresses.

## Synopsis

[Miscellaneous]      LicenseAltHeaders=*n*

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

## Description

For client IP addresses, by default, InterSystems IRIS uses the `remote_addr` HTTP header. If `LicenseAltHeaders` is true, then InterSystems IRIS instead uses the following headers (as needed, in the order given): `HTTP_FORWARDED`, or `HTTP_X_FORWARDED_FOR`, or `REMOTE_ADDR`.

## Changing This Parameter

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

Instead of using the Management Portal, you can change `LicenseAltHeaders` in the `Config.Miscellaneous` class (as described in the class reference) or by [editing the CPF](#) with a text editor.

---

# LineRecall

---

Allow command line recall for READ commands.

## Synopsis

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

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

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

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

# ListFormat

---

Specify the compression format for values in a list.

## Synopsis

[Miscellaneous]      ListFormat=*n*

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

## Description

**Important:**      This parameter has been retained for compatibility, but should not be used when building new applications.

ListFormat determines which values should be compressed within a list. The possible options for ListFormat are:

- 0 — no compression in a list
- 1 — [\\$DOUBLE](#) (IEEE) values in a list are compressed
- 2 — Unicode strings in a list are compressed
- 3 — Both [\\$DOUBLE](#) and Unicode strings in a list are compressed

**Note:**      If using lists with external clients (Java, C#, etc), ensure that the external client supports the compressed list format.

## Changing This Parameter

On the **Compatibility** page of the Management Portal (**System Administration > Configuration > Additional Settings > Compatibility**), in the **ListFormat** row, click **Edit**. Enter the desired value for this setting.

Instead of using the Management Portal, you can change ListFormat in the Config.Miscellaneous class (as described in the class reference) or by [editing the CPF](#) with a text editor.

## See Also

- [\\$DOUBLE](#) function
- [\\$LIST](#) function

# LogRollback

Allow logging for transaction rollbacks.

## Synopsis

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

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

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

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

## MVDefined

---

Not in use.

### Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

Not in use.



# NodeNameInPid

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

## Synopsis

```
[Miscellaneous]      NodeNameInPid=n
```

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

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

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

# NullSubscripts

---

Allow null subscripts on global references.

## Synopsis

[Miscellaneous]      NullSubscripts=*n*

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

## Description

**Important:**      This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

To change this parameter for a single process only (as opposed to instance-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.

## Synopsis

[Miscellaneous]      OldZU5=*n*

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

## Description

**Important:**      This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

To change this parameter for a single process only (as opposed to instance-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.

## Synopsis

[Miscellaneous]      `OpenMode=n`

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

## Description

**Important:**      This parameter has been retained for compatibility, but should not be used when building new applications.

`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](#) with a text editor.

To change this parameter for a single process only (as opposed to instance-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.

## Synopsis

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

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

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

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 instance, 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 instance, 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](#) with a text editor.

To change this parameter for a single process only (as opposed to instance-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.

## Synopsis

[Miscellaneous]      RefInKind=*n*

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

## Description

**Important:**      This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

**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 instance-wide), use the **RefInKind()** method of the %SYSTEM.Process class. See the class reference for details.

# ScientificNotation

---

Allow lowercase “e” as scientific notation symbol instance-wide.

## Synopsis

```
[Miscellaneous]      ScientificNotation=n
```

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

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

When `ScientificNotation` is enabled ( $n = 1$ ), InterSystems IRIS® data platform uses the lowercase “e” as scientific notation symbol instance-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](#) with a text editor.

To change this parameter for a single process only (as opposed to instance-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.

## Synopsis

[Miscellaneous]    `SetZEOF=n`

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

## Description

**Important:**    This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

To change this parameter for a single process only (as opposed to instance-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.

## Synopsis

```
[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](#).

## 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](#) with a text editor.

## StopID

---

Not in use.

### Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

Not in use.

# SwitchOSDir

Disallow switching current working directories when changing namespaces.

## Synopsis

```
[Miscellaneous]    SwitchOSDir=n
```

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

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

To change this parameter for a single process only (as opposed to instance-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.

## Synopsis

[Miscellaneous]      `SynchCommit=n`

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

## Description

**Important:**      This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

To change this parameter for a single process only (as opposed to instance-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.

## Synopsis

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

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

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor

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

# TruncateOverflow

---

Suppress the <MAXNUMBER> error on numeric overflow.

## Synopsis

[Miscellaneous]      `TruncateOverflow=n`

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

## Description

**Important:**      This parameter has been retained for compatibility, but should not be used when building new applications.

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 [Extremely Large Numbers](#).

## 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](#) with a text editor

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

## See Also

- [\\$DOUBLE](#) function

---

# Undefined

---

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

## Synopsis

```
[Miscellaneous]      Undefined=n
```

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

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor

To change this parameter for a single process only (as opposed to instance-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.

## Synopsis

[Miscellaneous]      UseNagleAlgorithm=*n*

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

## Description

**Important:**      This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.



# ViewPastData

---

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

## Synopsis

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

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

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

# ZDateNull

---

Specify the \$ZDATE response to an invalid value.

## Synopsis

[Miscellaneous]      ZDateNull=*n*

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

## Description

**Important:**      This parameter has been retained for compatibility, but should not be used when building new applications.

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](#) with a text editor.

To change this parameter for a single process only (as opposed to instance-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.

## Synopsis

```
[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**).

## 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 [Editing the Active CPF](#)).

## See Also

- [Monitoring InterSystems IRIS Using SNMP](#).

# [Namespaces]

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

# Namespace

Define InterSystems IRIS® data platform namespaces.

## Synopsis

```
[Namespaces]      Name=globals,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 instance 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 IRISYS 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 IRISYS (supplied by InterSystems IRIS).
- Routines and classes starting with %Z\* and %z\* come from IRISYS (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 IRISYS 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=IRISYS
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 settings on the **SQL** page of the Management Portal (**System Administration > Configuration > SQL and Object Settings > SQL**). The bottom of this page contains a list of parameters found in the [SQL] section of the CPF.

The **SQL** page is divided into the **SQL tab** and the **SQL Shell tab**. The **SQL tab** allows you to configure various SQL settings, which correspond to CPF parameters as shown in the table below. The **SQL Shell tab** options are described in [Configuring the SQL Shell](#).

**Table X-1: SQL Tab**

SQL Tab Setting	Equivalent CPF Parameter
Retain cached query source	<a href="#">SaveMAC</a>
Default time precision for GETDATE(), CURRENT_TIME, and CURRENT_TIMESTAMP	<a href="#">TimePrecision</a>
Lock escalation threshold	<a href="#">LockThreshold</a>
TO_DATE default format	<a href="#">TODATEDefaultFormat</a>
Default length for VARCHAR	<a href="#">ODBCVarcharMaxlen</a>
Default schema	<a href="#">DefaultSchema</a>
Execute queries in a single process	<a href="#">AutoParallel</a>
Define primary key as ID key for tables created via DDL	<a href="#">IdKey</a>
Ignore redundant DDL statements	Sets the <code>DDLNO*</code> parameters.
GROUP BY and DISTINCT queries must produce original values	<a href="#">FastDistinct</a>
Turn off Adaptive Mode to disable run time plan choice and automatic tuning	<a href="#">AdaptiveMode</a>
Turn on parameter sampling to sample the parameter value for query execution	<a href="#">ParameterSampling</a>
Lock timeout (seconds)	<a href="#">LockTimeout</a>
TCP keepalive for client connections (seconds)	<a href="#">TCPKeepAlive</a>
Client maximum idle time (seconds)	<a href="#">ClientMaxIdleTime</a>

---

# ANSIPrecedence

---

Specify operator precedence for SQL queries.

## Synopsis

```
[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 an instance-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, see [SQL Operator Precedence](#).

## Changing This Parameter

To set the desired value for `ANSIPrecedence` from the Terminal, use the `SetOption(“ANSIPrecedence”)` method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).

## See Also

- [Arithmetic Operators and Functions](#)

# AdaptiveMode

Enable Adaptive Mode performance optimization options.

## Synopsis

```
[SQL]      AdaptiveMode=n
```

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

## Description

When `AdaptiveMode` is enabled ( $n = 1$ ), InterSystems SQL uses Adaptive Mode, which enables automation and adaptive settings whenever possible. Adaptive Mode configures the following options:

- Disables freezing of existing query plans when you upgrade InterSystems IRIS® data platform to a new major version. Upon upgrade, non-frozen query plans are invalidated and replanned when called for the first time after upgrade. For more details, see [Frozen Plans](#).
- Enables parallel processing instance wide. For more details, see [Parallel Query Processing](#).
  - If the [ParallelProcessing](#) option is disabled (set to 0), Adaptive Mode overrides this setting and enables parallel processing.
- Enables automatic table tuning when an untuned table is first queried. With table tuning, InterSystems SQL gathers statistics from a table that it can use to optimize future queries. For more details, see [Tune Table](#).
- Enables Runtime Plan Choice. In doing so, InterSystems SQL takes the runtime values of query parameters into account to consider alternative plans, including in the presence of outliers. As a result, enabling this parameter overrides the older [BiasQueriesAsOutlier](#) and [RTPC](#) settings.

## Changing This Parameter

On the **SQL** page of the Management Portal (**System Administration** > **Configuration** > **SQL and Object Settings** > **SQL**), select **Turn off Adaptive Mode to disable run time plan choice and automatic tuning** to disable Adaptive Mode.

You can also change `AdaptiveMode` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

# AllowRowIDUpdate

---

Allow user to update RowID values.

## Synopsis

```
[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 [Editing the Active CPF](#)).

## See Also

- [RowID Field](#)

# AutoParallel

Allow parallel processing instance-wide.

## Synopsis

```
[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.

If [AdaptiveMode](#) is enabled (set to 1) and `AutoParallel` is disabled, then Adaptive Mode overrides the `AutoParallel` setting and enables parallel processing.

## 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 Terminal, use the `SetOption("AutoParallel")` method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).

# AutoParallelThreshold

---

Set the threshold for parallel processing.

## Synopsis

```
[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 Terminal, use the **SetOption**("AutoParallelThreshold") method of the %SYSTEM.SQL.Util 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 [Editing the Active CPF](#)).

# BiasQueriesAsOutlier

---

Set query optimization to biased toward outliers.

## Synopsis

```
[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, see [Tune Table](#).

`BiasQueriesAsOutlier` and [RTPC](#) cannot both be active at the same time. If both `BiasQueriesAsOutlier` and `RTPC` are set to 1, `RTPC` is activated and the `BiasQueriesAsOutlier` setting is ignored. When `RTPC` is set, InterSystems SQL determines whether to use outlier optimization on a per-query basis.

If [AdaptiveMode](#) is enabled (set to 1) and `BiasQueriesAsOutlier` is active and enabled, then Adaptive Mode overrides the `BiasQueriesAsOutlier` setting and disables outlier bias.

**Note:** Adaptive Mode replaces the earlier `BiasQueriesAsOutlier` mechanism. Please contact the WRC if you were previously using this setting and need assistance.

## Changing This Parameter

You can 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 [Editing the Active CPF](#)).

# ClientMaxIdleTime

---

## Synopsis

[SQL]      ClientMaxIdleTime=*n*

*n* is the number of seconds that the system allows for an idle connection between the client and the server before forcing a disconnection. The default value is 0.

## Description

When ClientMaxIdleTime is not 0, InterSystems SQL will wait the specified number of seconds before closing an idle connection between a client and a server. This timeout applies to all SQL clients that have been sitting idle, including JDBC, ODBC, and ADO.NET.

## Changing This Parameter

On the **SQL** page of the Management Portal (**System Administration > Configuration > SQL and Object Settings > SQL**), enter a value for the **Client maximum idle time (seconds)** setting.

You can change ClientMaxIdleTime by using SetOption() or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)). In addition, you can change the parameter on the **SQL** page of the Management Portal (**System Administration > Configuration > SQL and Object Settings > SQL**) by entering a value for the **Client maximum idle time (seconds)** setting.



---

# Comment

---

Retain embedded SQL statements as comments in source code.

## Synopsis

```
[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 Terminal, use the **SetOption("RetainSQL")** method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).

# DBMSSecurity

---

Enable SQL security.

## Synopsis

```
[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 Terminal, use the `SetOption("SQLSecurity")` method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).

# DDLDefineBitmapExtent

---

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

## Synopsis

```
[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 Terminal, use the **SetOption("DDLDefineBitmapExtent")** method of the %SYSTEM.SQL.Util 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 [Editing the Active CPF](#)).

# DDLFinal

---

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

## Synopsis

```
[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 Terminal, use the **SetOption("DDLFinal")** method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).

# DDLNo201

---

Suppress error upon CREATE of a previously existing table.

## Synopsis

[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.

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 [Editing the Active CPF](#)).

## DDLNo30

---

Suppress error upon DROP of a nonexistent table.

### Synopsis

```
[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.

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 [Editing the Active CPF](#)).

# DDLNo307

---

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

## Synopsis

[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.

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 [Editing the Active CPF](#)).

## DDLNo311

---

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

### Synopsis

```
[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.

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 [Editing the Active CPF](#)).



# DDLNo315

---

Suppress error upon DROP of a nonexistent constraint.

## Synopsis

[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.

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 [Editing the Active CPF](#)).

## DDLNo324

---

Suppress error upon CREATE of a previously existing index.

### Synopsis

[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.

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 [Editing the Active CPF](#)).

# DDLNo333

---

Suppress error upon DROP of a nonexistent index.

## Synopsis

[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.

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 [Editing the Active CPF](#)).

## DDLSQLOnlyCompile

---

Enable an SQL-only compile.

### Synopsis

```
[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 [Editing the Active CPF](#)).

# DDLUseExtentSet

---

Allow hashed names for globals that store index data of tables created by a DDL statement.

## Synopsis

```
[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 Terminal, use the **SetOption**("DDLUseExtentSet") method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).

## DDLUseSequence

---

Specify the function a table created by a DDL statement uses for ID assignment.

### Synopsis

```
[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 [\\$INCREMENT](#) or [\\$SEQUENCE](#).

### Changing This Parameter

To set the desired value for DDLUseSequence from the Terminal, use the **SetOption("DDLUseSequence")** method of the %SYSTEM.SQL.Util 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 [Editing the Active CPF](#)).

# DefaultSchema

Set the default SQL schema name.

## Synopsis

```
[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.

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 [Editing the Active CPF](#)).

---

## DelimitedIds

---

Enable interpreting double-quoted strings as delimited identifiers.

### Synopsis

```
[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. Also see [SQL Identifiers](#).

### Changing This Parameter

To set the desired value for `DelimitedIds` from the Terminal, use the `SetOption("DelimitedIdentifiers")` method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).



---

# DropDelete

---

Specify whether DROP TABLE deletes the table's data in addition to the table.

## Synopsis

```
[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 Terminal, use the **SetOption("DDLDropTabDelData")** method of the %SYSTEM.SQL.Util 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 [Editing the Active CPF](#)).

# ECPSync

---

Ensure that the server and client cache are in sync.

## Synopsis

```
[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, see [Queries and ECP](#).

## Changing This Parameter

To set the desired value for `ECPSync` from the InterSystems Terminal, use the **SetOption**("ECPSync") method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).

---

# ExtrinsicFunctions

---

Enable extrinsic functions in SQL statements.

## Synopsis

```
[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 Terminal, use the `SetOption("AllowExtrinsicFunctions")` method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).

# FastDistinct

---

Allow SQL DISTINCT optimization.

## Synopsis

```
[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 indexes (if indexes 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 Terminal, use the `SetOption("FastDistinct")` method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).

# IdKey

Set primary key constraint behavior.

## Synopsis

```
[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 Terminal, use the `SetOption("DDLPrimaryKeyNotIDKey")` method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).

---

# IdTrxFrom

---

Define the “From” list of characters for Identifier Translation.

## Synopsis

```
[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 [Identifiers](#). Also see the parameter [IdTrxTo](#).

## Changing This Parameter

To set the desired value for IdTrxFrom from the Terminal, use the **SetDDLIdentifierTranslations()** method of the %SYSTEM.SQL.Util 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 [Editing the Active CPF](#)).

# IdTrxTo

---

Define the “To” list of characters for Identifier Translation.

## Synopsis

```
[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 [Identifiers](#). Also see [IdTrxFrom](#).

## Changing This Parameter

To set the desired value for IdTrxTo from the Terminal, use the **SetDDLIdentifierTranslations()** method of the %SYSTEM.SQL.Util 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 [Editing the Active CPF](#)).

---

# LockThreshold

---

Set the SQL table-level lock threshold.

## Synopsis

```
[SQL]      LockThreshold=n
```

*n* is any nonnegative integer. The default value is 1000.

## Description

The `LockThreshold` parameter is the automatic lock escalation 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.

Consider this example: a process starts a transaction that inserts 2000 rows, where `LockThreshold` is set to 1000. After the 1001<sup>st</sup> row is inserted, the process attempts to acquire a table-level lock rather than continue to lock individual rows. This reduces the total number of locks to prevent the lock table from becoming too full.

Automatic lock escalation is intended to prevent overflow of the lock table. For further details, see [Modify Transaction Lock Threshold](#).

## 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 Terminal, use the `SetOption(“LockThreshold”)` method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).

## See Also

- The `gmheap` and `locksiz` parameters (to increase the size of the lock table).
- [The Lock Table](#)
- [Modify Transaction Lock Threshold](#) in *InterSystems SQL Reference*.



---

# LockTimeout

---

Set the SQL lock timeout.

## Synopsis

```
[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 Terminal, use the **SetOption(“LockTimeout”)** method of the %SYSTEM.SQL.Util 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 [Editing the Active CPF](#)).

# ODBCVarcharMaxlen

---

Set the MaxLen for ODBC fields of type VarChar.

## Synopsis

```
[SQL]      ODBCVarcharMaxlen=n
```

*n* is any nonnegative integer. The maximum value is the [maximum string length](#). 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 [Editing the Active CPF](#)).

# ParameterSampling

---

Specify whether parameter sampling is enabled or not.

## Synopsis

```
[SQL]      ParameterSampling=n
```

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

## Description

When `ParameterSampling` is enabled ( $n = 1$ ), InterSystems SQL will save the complete set of query parameters when statements are prepared. This data is projected to `INFORMATION_SCHEMA.STATEMENT_PARAMETER_STATS`, which includes information on the efficiency of the query. Enabling parameter sampling can be useful for evaluating the efficacy of a schema, especially when testing a new schema.

## Changing This Parameter

On the **SQL** page of the Management Portal (**System Administration** > **Configuration** > **SQL and Object Settings** > **SQL**), select **Turn on parameter sampling to sample the parameter value for query execution** to enable `ParameterSampling`.

To set the desired value for `ParameterSampling` from the InterSystems Terminal, use the **SetOption("ParameterSampling")** method of the `%SYSTEM.SQL.Util` class. See the class reference for details.

You can also change `ParameterSampling` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

# QueryProcedures

---

Specify whether all class queries project as SQL Stored Procedures.

## Synopsis

```
[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 n instance restart to make it active.

## Changing This Parameter

To set the desired value for `QueryProcedures` from the Terminal, use the **`SetOption("QueryProcedures")`** method of the `%SYSTEM.SQL.Util` class. See the class reference for details.

You can also change `QueryProcedures` with the `Config.SQL` class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

---

# RTPC

---

Enable Runtime Plan Choice (RTPC) query optimization.

## Synopsis

[SQL]      RTPC=*n*

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

## Description

When RTPC is enabled ( $n = 1$ ), InterSystems SQL performs extensive optimization of the query based on query input values. For example, RTPC does the following:

- Scans for outlier values and optimizes queries based on outlier information.
- Efficiently estimates the selectivity of range conditions based on more detailed table statistics.
- Evaluates explicit and implicit truth value conditions.
- Efficiently estimates the selectivity of list conditions.

For more information about RTPC query optimization, see [Using Runtime Plan Choice](#).

RTPC and [BiasQueriesAsOutlier](#) cannot both be active at the same time. If both RTPC and `BiasQueriesAsOutlier` are set to 1, RTPC is activated and the `BiasQueriesAsOutlier` setting is ignored. When RTPC is set, InterSystems SQL determines whether to use outlier optimization on a per-query basis.

If [AdaptiveMode](#) is enabled (set to 1) and RTPC is disabled, then Adaptive Mode overrides the RTPC setting and enables Runtime Plan Choice.

## Changing This Parameter

To set the desired value for RTPC from the Terminal, use the `SetOption("RTPC")` method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).

---

# ReferentialChecks

---

Enable foreign key constraint validation.

## Synopsis

```
[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 Terminal, use the `SetOption("FilerRefIntegrity")` method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)).

---

# SaveMAC

---

Save the source code for cached query routines.

## Synopsis

[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 Terminal, use the **SetOption("CachedQuerySaveSource")** method of the %SYSTEM.SQL.Util 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 [Editing the Active CPF](#)).

---

# TCPKeepAlive

---

Set the number of seconds between keep-alive messages.

## Synopsis

```
[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 instance uses the operating system default.

For further details, see [TCP Client/Server Communication](#).

## 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 TCPKeepAlive from the InterSystems Terminal, use the **SetOption**("TCPKeepAlive") method of the %SYSTEM.SQL.Util class. See the class reference for details.

You can also change TCPKeepAlive with the Config.SQL class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

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](#).



# TODATEDefaultFormat

Set the default date format for the SQL TO\_DATE() function.

## Synopsis

```
[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](#) argument description of the **TO\_DATE** reference page.

## 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 Terminal, use the **SetOption("ToDateDefaultFormat")** method of the %SYSTEM.SQL.Util 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 [Editing the Active CPF](#)).

---

# TimePrecision

---

Set the default time precision for SQL scalar time functions.

## Synopsis

```
[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 Terminal, use the **SetOption(“DefaultTimePrecision”)** method of the `%SYSTEM.SQL.Util` 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 [Editing the Active CPF](#)), or by using the **SET OPTION** command.

# [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.

## Synopsis

```
[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](#).

```
[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 [Editing the Active CPF](#)).

## See Also

- [User Datatypes](#)
- [Data Types](#)

# [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.

## Synopsis

```
[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 [Editing the Active CPF](#)).

## See Also

- [System Datatypes](#)
- [Data Types](#)

# [Startup]

This topic describes the **Startup Settings** parameters found in the [Startup] section of the CPF. Many of these, including the mirror-related and sharded cluster-related parameters, are in deployment with configuration merge.

# CallinHalt

---

Allow custom routines during callin close.

## Synopsis

```
[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 [Editing the Active CPF](#)).

## 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 instance 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](#).



# CallinStart

---

Allow custom routines during callin initialization.

## Synopsis

```
[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 [Editing the Active CPF](#)).

## 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 instance 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](#).

# CliSysName

---

Set the node name for the local system.

## Synopsis

```
[Startup]      CliSysName=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 [Editing the Active CPF](#)).

# DBSizesAllowed

Specify possible database block sizes when creating a database.

## Synopsis

```
[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 [Configuring Databases](#).

## 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 [Editing the Active CPF file](#)).

## See Also

- [globals](#) parameter in the [Config] section of this reference
- [Configuring Databases](#)
- [Allocating Memory to the Database and Routine Caches](#)
- [Large Block Size Considerations](#)

# DefaultPort

---

Set the port number for the InterSystems IRIS® data platform superserver.

## Synopsis

```
[Startup]      DefaultPort=n
```

*n* is a valid port number. The default is 1972.

## Description

DefaultPort is the port number for the InterSystems IRIS superserver. The superserver listens on a specified port (1972 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 1972. If that port is in use by another InterSystems IRIS instance on the same system, then InterSystems IRIS sets this value to 51773 or the next available subsequent 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 [Editing the Active CPF](#)).

---

# DefaultPortBindAddress

---

Set the IP address for the InterSystems IRIS® data platform superserver to bind to.

## Synopsis

```
[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 [Editing the Active CPF](#)).

# EnableSharding

Enable the sharding service for this instance.

## Synopsis

```
[Startup]      EnableSharding=n
```

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

## Description

When this parameter is enabled ( $n = 1$ ), InterSystems IRIS® data platform enables the Sharding service (**%Service\_Sharding**) during instance startup. When **EnableSharding** is disabled ( $n = 0$ ), **%Service\_Sharding** is unchanged during instance startup. This means **EnableSharding** can only be used to enable the Sharding service, *not* to disable it.

**EnableSharding** is not in the default CPF, but can be included as a parameter in a [configuration merge file](#). To directly enable **%Service\_Sharding** without the **EnableSharding** parameter, use the **Services** page of the Management Portal (**System Administration > Security > Services**).

## Changing This Parameter

**EnableSharding** is not in the default CPF. You can manually add this parameter to the [Startup] section of the CPF using a text editor (as described in [Editing the Active CPF](#)).

After adding **EnableSharding** and restarting InterSystems IRIS, you can change the parameter using a text editor, the Config.Startup class (as described in the class reference), or the **Startup** page of the Management Portal (**System Administration > Configuration > Additional Settings > Startup**).

## See Also

- [Services](#).

# EnableVSSBackup

---

Allow Volume Shadow Copy Service (VSS) backup. Windows systems only.

## Synopsis

```
[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 [Backup Strategies](#) 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 [Editing the Active CPF](#)).

# EnsembleAutoStart

---

Allow productions to auto-start when InterSystems IRIS® data platform starts.

## Synopsis

```
[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 [Starting and Stopping 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 [Editing the Active CPF](#)).



---

# ErrorPurge

---

Set the number of days to keep error globals.

## Synopsis

```
[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 [Editing the Active CPF](#)).

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.

## Synopsis

```
[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.

**Note:** Enabling `FIPSMoDe` only affects encrypted databases. To encrypt a database, choose the **Encrypt Database?** option during the [database creation process](#).

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 [Editing the Active CPF](#)).

---

# IPv6

---

Allow InterSystems IRIS® data platform to accept IPv6 addresses.

## Synopsis

```
[Startup]      IPv6=n
```

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

## Description

IPv6 controls whether your instance is operating in an IPv6 (Internet Protocol Version 6) network, with IPv6 addresses. For more information, see [IPv6 Support](#).

## 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 [Editing the Active CPF](#)).

# JobHalt

Allow custom routines during background process (job) shutdown.

## Synopsis

```
[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 [Editing the Active CPF](#)).

## 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 instance 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](#).

# JobServers

Set the number of job servers.

## Synopsis

```
[Startup]      JobServers=n
```

*n* is an integer in the range 0—2000. The default value is 0.

## Description

The `JobServers` parameter sets both the target number of available job servers you want to maintain and the number that are created at startup. It is *not* the total number of job servers. When the number of available job servers falls below the target number, more job servers are created in order to maintain the number of available job servers.

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.

You can use the `^JOBSRV` routine to display information about job servers including the configuration and the number of job servers.

## System Behavior

When job servers are enabled, the system runs a monitor process every 5 seconds to determine the number of available job servers. If the number of available job servers is *below* the target number, more job servers are created until the target is reached.

Every 3 minutes, the monitor also checks if the number of available job servers *exceeds* the target number. If it does, it stops a single job server. However, the total number of running job servers will never be decreased to fewer than the number set by the `JobServers` parameter.

If the `JobServers` parameter is changed to 0 (disabling job servers), the monitor immediately stops all available job servers and continues to stop utilized job servers as they become available.

When utilized job servers complete their work, they are automatically returned to the available pool.

## Determining the Target Number

The system will maintain a number of available job servers based on an effective target number. The effective target number is determined by whichever is smaller: the parameter or the dynamic target number determined by the total number of job servers (see the table below). For example, if you set `JobServers=7`, the effective target number will be 5 when there are fewer than 20 total job servers and 7 when there are 20 or more. The [examples](#) below illustrate this in more detail.

Total Number of Job Servers	Dynamic Target Number
1 – 4	determined by the parameter
5 – 19	5
20 – 99	10
100+	20

## Examples

### Parameter: JobServers=4

Every time a new process starts, it uses one of the available job servers, so the system creates a new job server.

	Parameter	Dynamic Target	Effective Target	Available	Utilized	Total
On Startup	4	N/A	4	4	0	4
1 process	4	N/A	4	4	1	5
5 processes	4	N/A	4	4	5	9

### Parameter: JobServers=12

The system initially starts with 12 available job servers, but an effective target of 5. When the number of available job servers drops below 5, more are created to maintain the effective target number.

When 15 of the job servers are utilized, the system reaches 20 total job servers (15 utilized and 5 available). This causes the effective target to increase to 10, so the system creates 5 additional job servers. The system now has 25 total job servers (15 utilized and 10 available).

When 90 of the job servers are utilized, the system reaches 100 total job servers (90 utilized and 10 available). This causes the dynamic target to increase to 20. Since the parameter is not less than the dynamic target, it becomes the effective target. The system only creates 2 additional job servers in order to meet the effect target of 12.

	Parameter	Dynamic Target	Effective Target	Available	Utilized	Total
On Startup	12	5	5	12	0	12
10 processes	12	5	5	5	10	15
15 processes	12	5 10	5 10	5 10	15	20 25
30 processes	12	10	10	10	30	40
90 processes	12	10 100	10 12	10 12	90	100 102
150 processes	12	100	12	12	100	112

### Parameter: JobServers=100

The system initially starts with 100 available job servers, but an effective target of 20. Once the number of available job servers is reduced to the effective target number (as they are utilized), the system begins creating new job servers to maintain the number available.

	Parameter	Dynamic Target	Effective Target	Available	Utilized	Total
<i>On Startup</i>	100	20	20	100	0	100
<i>50 processes</i>	100	20	20	50	50	100
<i>80 processes</i>	100	20	20	20	80	100
<i>90 processes</i>	100	20	20	20	90	110

## 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 [Editing the Active CPF](#)).

# JobStart

Allow custom routines during background process (job) startup.

## Synopsis

```
[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 [Editing the Active CPF](#)).

## 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 instance 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](#).



---

# LicenseID

---

Allow instance to request a key from the license server.

## Synopsis

```
[Startup]      LicenseID=n
```

*n* is the name of a license key in the target .key file. By default, no key is specified.

## Description

If InterSystems IRIS® data platform does not detect a local iris.key file at instance startup, it 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 [Managing InterSystems IRIS Licensing](#).

## 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 [Editing the Active CPF](#)).

# MaxConsoleLogSize

---

Set the maximum size of the messages.log file.

## Synopsis

```
[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 instance 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 [Editing the Active CPF](#)).

# MaxIRISTempSizeAtStart

Set the maximum size of the IRISTEMP database on restart.

## Synopsis

```
[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 physical size in megabytes of the IRISTEMP database during instance startup. When the instance restarts, InterSystems IRIS® data platform truncates the IRISTEMP database to the size that MaxIRISTempSizeAtStart specifies. If this parameter is set to 0, the IRISTEMP database is not truncated on instance restart.

**Note:** InterSystems IRIS initializes a maximum of 240 MB in the IRISTEMP database at instance startup. This may be significantly less than the physical size of the database's IRIS.DAT file.

You can review the IRISTEMP [free space information](#) to see the logical size of IRISTEMP, and manually [truncate the database](#) from the Management Portal if IRISTEMP grows too large.

## 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 [Editing the Active CPF](#)).

# PasswordHash

Set the default password for the predefined user accounts using a cryptographic hash of the password and its salt.

## Synopsis

```
[Startup] PasswordHash=a,b[,c,d]
```

*a* is a hashed password. *b* is its salt, *c* and *d* optionally indicate the work factor and algorithm used to hash the password.

## Description

The primary use for `PasswordHash` is to set the default password for the [predefined user accounts](#) in automated deployments. `PasswordHash` specifies a hashed password and its salt, and optionally the work factor (default 10000) and the hashing algorithm (default SHA512) used to hash the password. When it is used in a CPF merge file, on instance startup the stored password hash for each enabled user account with at least one assigned role — that is, all of the predefined accounts except **CSPSystem**, which does not have an assigned role — is set to the value of `PasswordHash`. If `PasswordHash` is empty, the CPF merge operation ignores it.

**Important:** The `PasswordHash` property can be used just once on any given InterSystems IRIS instance, and only if the default password has not yet been changed for any of the predefined accounts. Because allowing the default password to remain unchanged following deployment is a serious security risk, the `PasswordHash` setting should be used in a configuration merge operation to change the default password during deployment and not later. (For information on how to change an individual user's password, see [Edit an Existing User Account](#).)

`PasswordHash` is not recommended for changing passwords on a deployed instance.

## Example

For details about hashing a password and an example of using `PasswordHash` when deploying a container, see [Authentication and Passwords](#).

## See Also

- [Authentication and Passwords](#)
- [Instance Authentication](#)
- [Editing an Existing User Account](#)

# ProcessHalt

Allow custom routines during foreground process shutdown.

## Synopsis

```
[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 [Editing the Active CPF](#)).

## 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 instance 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](#).

---

# ProcessStart

---

Allow custom routines during foreground process startup.

## Synopsis

```
[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 [Editing the Active CPF](#)).

## 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 instance 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](#).

---

# ShutdownTimeout

---

Set the number of seconds InterSystems IRIS® data platform should wait until forcing a shutdown.

## Synopsis

```
[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 [Editing the Active CPF](#)).

---

# SystemHalt

---

Allow custom routines during instance shutdown.

## Synopsis

```
[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 instance 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 [Editing the Active CPF](#)).

## 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 instance 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](#).



# SystemMode

Specify a label that appears in the Management Portal header.

## Synopsis

```
[Startup]      SystemMode=n
```

*n* is an alphanumeric string. The maximum length is 32 characters. The default is an empty string.

## Description

`SystemMode` defines a label that appears in the [Management Portal header](#). This label can be used to easily identify an instance of InterSystems IRIS® data platform.

You may input any label, though some `SystemMode` values receive special treatment in the Management Portal. These values are:

- `LIVE` — Translates to **Live System** with red text and a red border.
- `TEST` — Translates to **Test System**.
- `DEVELOPMENT` — Translates to **Development System**.
- `FAILOVER` — Translates to **Failover System**.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration** > **Configuration** > **Additional Settings** > **Startup**), in the **SystemMode** row, select **Edit**. Enter the desired label in the **SystemMode** text box, then click **Save**.

Instead of using the Management Portal, you can change `SystemMode` in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

---

# SystemStart

---

Allow custom routines during instance startup.

## Synopsis

```
[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 instance 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 [Editing the Active CPF](#)).

## 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 instance 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](#).

# TempDirectory

Specify the subdirectory for temporary files.

## Synopsis

```
[Startup]    TempDirectory=n
```

*n* is an absolute or relative directory pathname. Provided that the value of TempDirectory is a valid pathname, there is no maximum length. 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 instance 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 [Editing the Active CPF](#)).

# TerminalPrompt

Define the format of the terminal prompt.

## Synopsis

```
[Startup]    TerminalPrompt=n
```

*n* is a string of comma-separated values (0–8) that set the Terminal prompt. The default is 8 , 2.

## Description

`TerminalPrompt` configures the information displayed in the Terminal prompt for the instance. The order of the values in the string determines the order the information appears in the prompt. For example, the string 2 , 1 produces a terminal prompt of `%SYS:HostName>`.

The values are coded as follows:

- 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 `$NAMESPACE` special variable. It can be an explicit namespace name or an implied namespace name.
- 3 - Config name. The name of your instance. For example, `IRIS2>`. This is the same for all of your terminal processes.
- 4 - Current time, expressed as local time in 24-hour format with whole seconds. For example, `15:59:36>`. This is the static time value for when the prompt was returned. This value changes for each prompt.
- 5 - The Process ID for your terminal. For example, `2336>`. This is different for each terminal process. This value can also be returned from the `$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>`.

`TerminalPrompt` cannot be left empty. If the input string is invalid, the Terminal uses the default value for `TerminalPrompt: 8 , 2`.

## 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 [Editing the Active CPF](#)).

# WebServer

Allow the private web server to start, if one is present.

## Synopsis

**Important:** Versions of InterSystems IRIS® prior to 2023.2 included a private web server that served built-in web applications such as the Management Portal. Beginning with 2023.2, new installations of InterSystems IRIS no longer include a private web server. See [this section](#) of the *Web Gateway Guide* for more information.

```
[Startup]    WebServer=n
```

*n* is either 1 (true) or 0 (false). The default value is 0 for new installations, 1 for upgrades.

## Description

When `WebServer` is enabled ( $n = 1$ ), InterSystems IRIS attempts to start the Apache private web server upon startup.

## 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 the private web server.

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 [Editing the Active CPF](#)).

**Note:** A restart is required for this setting to take effect.

## See Also

- [WebServerName](#)
- [WebServerPort](#)
- [WebServerProtocol](#)
- [WebServerURLPrefix](#)

# WebServerName

---

Identify an IP address or a DNS name for the web server.

## Synopsis

[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

**Important:**     Note that this parameter is only used in conjunction with Studio, which has been removed as of the 2024.2 release. See [Deprecated and Discontinued Features](#) for more information.

WebServerName identifies the DNS name or the IP address of the web server that is configured for use with InterSystems IRIS® data platform tools. It is necessary to set this parameter only if you must enable InterSystems Studio.

## 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 [Editing the Active CPF](#)).

## See Also

- [WebServer](#)
- [WebServerPort](#)
- [WebServerProtocol](#)
- [WebServerURLPrefix](#)

# WebServerPort

---

Identify the web server port.

## Synopsis

```
[Startup]    WebServerPort=nnnnn
```

*n* is a valid port number. The default is 80.

## Description

**Important:** Note that this parameter is only used in conjunction with Studio, which has been removed as of the 2024.2 release. See [Deprecated and Discontinued Features](#) for more information.

WebServerPort identifies the port number of the web server that is configured for use with InterSystems IRIS® data platform tools. It is necessary to set this parameter only if you must enable InterSystems Studio.

## 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 [Editing the Active CPF](#)).

## See Also

- [WebServer](#)
- [WebServerName](#)
- [WebServerProtocol](#)
- [WebServerURLPrefix](#)

# WebServerProtocol

---

Enable InterSystems IRIS to receive HTTPS connections over a custom port.

## Synopsis

```
[Startup]    WebServerProtocol=protocol
```

*protocol* is either http (the default) or https.

## Description

**Important:** Note that this parameter is only used in conjunction with Studio, which has been removed as of the 2024.2 release. See [Deprecated and Discontinued Features](#) for more information.

When `WebServerPort` is set to 443 (the standard port for HTTPS), Studio automatically uses HTTPS. However, if it is set to some other value, `WebServerProtocol` must be set to `https` to enable Studio to communicate over HTTPS. It is necessary to set this parameter only if you must enable InterSystems Studio.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration** > **Configuration** > **Additional Settings** > **Startup**), in the **WebServerProtocol** row, select **Edit**. Enter `https` or `http`.

Instead of using the Management Portal, you can change `WebServerProtocol` in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

## See Also

- [WebServer](#)
- [WebServerName](#)
- [WebServerPort](#)
- [WebServerURLPrefix](#)



# WebServerSSLConfiguration

Specify the default TLS configuration for %Net.HttpRequest objects created by CSP templates implemented in Studio.

## Synopsis

```
[Startup]    WebServerSSLConfiguration=configuration
```

*configuration* is the location of a TLS configuration

## Description

**Important:** This parameter has been retained for compatibility, but should not be used when building new applications.

Deprecated. Must be specified when using Studio to create a web application from a CSP template on an instance with an active private web server that accepts only HTTPS connections.

## Changing This Parameter

On the **Startup** page of the Management Portal (**System Administration** > **Configuration** > **Additional Settings** > **Startup**), in the **WebServerProtocol** row, select **Edit**. Enter the location of the TLS configuration.

Instead of using the Management Portal, you can change `WebServerSSLConfiguration` in the `Config.Startup` class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

## See Also

- [WebServer](#)
- [WebServerName](#)
- [WebServerPort](#)
- [WebServerProtocol](#)
- [WebServerURLPrefix](#)

# WebServerURLPrefix

Identify the instance prefix that the web server uses to access the InterSystems IRIS® data platform.

## Synopsis

```
[Startup]      WebServerURLPrefix=n
```

*n* is an alphanumeric string to be used in a URL. As a guideline, WebServerURLPrefix should be shorter than 80 characters. The default is the instance name in all lowercase characters.

## Description

**Important:** Note that this parameter is only used in conjunction with Studio, which has been removed as of the 2024.2 release. See [Deprecated and Discontinued Features](#) for more information.

WebServerURLPrefix is used by Studio when constructing URLs. It is necessary to set this parameter only if you must enable InterSystems Studio. This should match the [CSP Server Instance](#) setting.

## 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 [Editing the Active CPF](#)).

## See Also

- [WebServer](#)
- [WebServerName](#)
- [WebServerPort](#)
- [WebServerProtocol](#)

---

# ZSTU

---

Allow the user-defined startup to run.

## Synopsis

```
[Startup]      ZSTU=n
```

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

## Description

**Important:** This parameter has been retained for compatibility, but the ^ZSTU routine is no longer recommended for use; use the [^%ZSTART routine](#) instead.

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 [Editing the Active CPF](#)).



# [Telnet]

This topic describes the parameters found in the [Telnet] section of the CPF. These settings only apply 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.

You can also find these settings on the **Telnet Settings** page of the Management Portal (**System Administration > Configuration > Device Settings > Telnet Settings**).

# DNSLookup

---

Allow DNS lookup of the Telnet client address. Windows systems only.

## Synopsis

```
[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 [Editing the Active CPF](#)).

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

---

# Port

---

Set the Telnet port number. Windows systems only.

## Synopsis

```
[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 [Editing the Active CPF](#)).





# [WorkQueues]

This topic describes the work queue manager categories found in the [WorkQueues] section of the CPF. You may create a new category using the same syntax as an existing category. For more information about the Work Queue Manager, see [Using the Work Queue Manager](#).

## Default

---

Define the “Default” work queue manager category.

### Synopsis

```
[WorkQueues]      Default=MaxActiveWorkers,DefaultWorkers,MaxWorkers
```

`Default` is the name of this work queue manager category. *MaxActiveWorkers*, *DefaultWorkers*, and *MaxWorkers* are three comma-separated integers. Any of these may be left blank or specified as 0 to use the default value. If using default values for *MaxActiveWorkers* and *MaxWorkers*, the only hard limit on the parameters is the number of *MaxActiveWorkers* in the work queue manager category. Specifying a value of -1 for *MaxActiveWorkers* allows for an unlimited number of active workers in the work queue manager category.

### Description

The [WorkQueues] section of the configuration parameter file (CPF) contains an entry for every work queue manager category. A category has up to three comma-separated properties. None of the arguments are required; if any are left blank, they are resolved as the default value. For more information, see [Managing Categories](#).

### Changing This Parameter

On the **Work Queue Manager Categories** page of the Management Portal (**System Administration > Configuration > System Configuration > WQM Categories**), click **Default** to edit this category. you can instead click **Create Category** to create a new WQM category.

Instead of using the Management Portal, you can change the `Default` category in the `Config.WorkQueues` class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

### See Also

- [Using the Work Queue Manager](#)
- [Creating a Work Queue Manager Category](#)

---

# SQL

---

Define the “SQL” work queue manager category.

## Synopsis

```
[WorkQueues]      SQL=MaxActiveWorkers,DefaultWorkers,MaxWorkers
```

SQL is the name of this work queue manager category. *MaxActiveWorkers*, *DefaultWorkers*, and *MaxWorkers* are three comma-separated integers. Any of these may be left blank or specified as 0 to use the default value. If using default values for *MaxActiveWorkers* and *MaxWorkers*, the only hard limit on the parameters is the number of *MaxActiveWorkers* in the work queue manager category. Specifying a value of -1 for *MaxActiveWorkers* allows for an unlimited number of active workers in the work queue manager category.

## Description

The [WorkQueues] section of the configuration parameter file (CPF) contains an entry for every work queue manager category. A category has up to three comma-separated properties. None of the arguments are required; if any are left blank, they are resolved as the default value. For more information, see [Managing Categories](#).

## Changing This Parameter

On the **Work Queue Manager Categories** page of the Management Portal (**System Administration > Configuration > System Configuration > WQM Categories**), click **SQL** to edit this category. you can instead click **Create Category** to create a new WQM category.

Instead of using the Management Portal, you can change the SQL category in the Config.WorkQueues class (as described in the class reference) or by editing the CPF in a text editor (as described in [Editing the Active CPF](#)).

## See Also

- [Using the Work Queue Manager](#)
- [Configuring Work Queue Manager Categories](#)

