



構成パラメータ・ファイル・リ ファレンス

Version 2024.1
2024-06-03

構成パラメータ・ファイル・リファレンス

InterSystems IRIS Data Platform Version 2024.1 2024-06-03

Copyright © 2024 InterSystems Corporation

All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, および TrakCare は、InterSystems Corporation の登録商標です。HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, および InterSystems TotalView™ For Asset Management は、InterSystems Corporation の商標です。TrakCare は、オーストラリアおよび EU における登録商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼動および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems Worldwide Response Center (WRC)

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: support@InterSystems.com

目次

1 構成パラメータ・ファイルの概要	1
1.1 CPF の概要	1
1.1.1 CPF の形式	1
1.1.2 CPF のサンプル	2
1.2 アクティブな CPF の編集	7
1.2.1 起動時の CPF の選択	7
1.3 構成マージ機能の使用	8
1.3.1 構成マージの例	9
1.4 構成のセキュリティ	9
1.5 パラメータの説明	10
2 [Actions]	11
[Archives]	13
Archives	14
[ComPorts]	17
COMn	18
[config]	23
LibPath	24
MaxServerConn	25
MaxServers	26
Path	27
PythonPath	28
PythonRuntimeLibrary	29
bbsiz	30
console	31
errlog	32
globals	33
gmheap	34
history	35
ijcbuff	36
ijcnum	37
jrnbufs	38
locksiz	39
memlock	40
netjob	42
nlstab	43
overview	44
pijdir	45
routines	46
udevtabsiz	48
wijdir	49
targwijsz	50
zfheap	51
[ConfigFile]	53
Version	54

[Databases]	55
Database	56
[Debug]	59
Dumpstyle	60
Semsperset	62
[Devices]	63
Devices	64
[DeviceSubTypes]	67
DeviceSubTypes	68
[ECP]	71
ClientReconnectDuration	72
ClientReconnectInterval	73
ServerTroubleDuration	74
[ECPServers]	75
ECPServers	76
[Gateways]	77
%DotNet Server	78
%IntegratedML Server	80
%JDBC Server	81
%Java Server	82
%Python Server	83
%R Server	84
%XSLT Server	85
リモートサーバ	86
[IO]	87
Other	88
File	89
Terminal	90
[Journal]	91
AlternateDirectory	92
ArchiveName	93
BackupsBeforePurge	94
CompressFiles	95
CurrentDirectory	96
DaysBeforePurge	97
FileSizeLimit	98
FreezeOnError	99
JournalFilePrefix	100
JournalcspSession	101
PurgeArchive	102
[LicenseServers]	103
LicenseServers	104
[Logging]	105
ChildProcessLaunchCommand	106
Enabled	107
Format	108

interval	109
Level	110
[Map]	111
Global	112
Package	114
Routine	115
[MapMirrors]	117
MapMirrors	118
[MirrorMember]	121
AgentAddress	122
AsyncMemberGUID	123
AsyncMemberType	124
AsyncUseSystemPurgeInterval	125
JoinMirror	126
SystemName	127
ValidatedMember	128
VirtualAddressInterface	129
[Mirrors]	131
Mirrors	132
[Miscellaneous]	135
AsyncDisconnectErr	136
AsynchError	137
BreakMode	138
CollectResourceStats	139
DisconnectErr	140
FileMode	141
GlobalKillEnabled	142
IEEEError	143
LicenseAltHeaders	144
LineRecall	145
ListFormat	146
LogRollback	147
MVDefined	148
NodeNameInPid	149
NullSubscripts	150
OldZU5	151
OpenMode	152
PopError	153
ReflnKind	154
ScientificNotation	155
SetZEOF	156
ShutDownLogErrors	157
StopID	158
SwitchOSdir	159
SynchCommit	160
TelnetNUL	161
TruncateOverflow	162
Undefined	163

UseNagleAlgorithm	164
ViewPastData	165
ZDateNull	166
[Monitor]	167
SNMPEntered	168
[Namespaces]	169
Namespace	170
[SQL]	173
ANSIPrecedence	174
AdaptiveMode	175
AllowRowIDUpdate	176
AutoParallel	177
AutoParallelThreshold	178
BiasQueriesAsOutlier	179
ClientMaxIdleTime	180
Comment	181
DBMSSecurity	182
DDLDefineBitmapExtent	183
DDLFinal	184
DDLNo201	185
DDLNo30	186
DDLNo307	187
DDLNo311	188
DDLNo315	189
DDLNo324	190
DDLNo333	191
DDLSQLOnlyCompile	192
DDLUseExtentSet	193
DDLUseSequence	194
DefaultSchema	195
DelimitedIds	196
DropDelete	197
ECPSync	198
ExtrinsicFunctions	199
FastDistinct	200
IdKey	201
IdTrxFrom	202
IdTrxTo	203
LockThreshold	204
LockTimeout	205
ODBCVarcharMaxlen	206
ParameterSampling	207
QueryProcedures	208
RTPC	209
ReferentialChecks	210
SaveMAC	211
TCPKeepAlive	212
TODATEDefaultFormat	213
TimePrecision	214

[SqlSysDatatypes]	215
システム・データ型	216
[SqlUserDatatypes]	217
ユーザ・データ型	218
[Startup]	219
CallinHalt	220
CallinStart	221
CliSysName	222
DBSizesAllowed	223
DefaultPort	224
DefaultPortBindAddress	225
EnableSharding	226
EnableVSSBackup	227
EnsembleAutoStart	228
ErrorPurge	229
FIPSMODE	230
IPv6	231
JobHalt	232
JobServers	233
JobStart	234
LicenseID	235
MaxConsoleLogSize	236
MaxIRISTempSizeAtStart	237
PasswordHash	238
ProcessHalt	239
ProcessStart	240
ShutdownTimeout	241
SystemHalt	242
SystemMode	243
SystemStart	244
TempDirectory	245
TerminalPrompt	246
WebServer	247
WebServerName	248
WebServerPort	249
WebServerProtocol	250
WebServerSSLConfiguration	251
WebServerURLPrefix	252
ZSTU	253
[Telnet]	255
DNSLookup	256
Port	257
[WorkQueues]	259
Default	260
SQL	261

テーブル一覧

テーブル W-1: SQL タブ	173
------------------------	-----

1

構成パラメータ・ファイルの概要

InterSystems IRIS® データ・プラットフォームは、起動時に構成パラメータ・ファイル (CPF) `iris.cpf` から構成値を読み取ります。このファイルには、InterSystems IRIS の特定の構成がインスタンスごとに定義されています。

このトピックでは、CPF の使用および編集の方法について説明します。このリファレンスの最初にある“[目次](#)”に、CPF の各パラメータをセクション別にソートして示しています。

注釈 このドキュメントおよび CPF そのものの中では、メモリを 2 の累乗として表します。例えば、1 キロバイト (KB) は 1024 バイトを意味し、1 メガバイト (MB) は 1024 KB を意味します。

1.1 CPF の概要

構成パラメータ・ファイル (CPF) は、InterSystems IRIS 構成を定義します。起動時に、InterSystems IRIS は CPF を読み取ってそのほとんどの設定の値を取得します。

既定の CPF である `iris.cpf` は、[インストール・ディレクトリ](#)にあります。“[アクティブな CPF の編集](#)”のセクションで説明されているように、CPF を編集する方法は複数あります。

InterSystems IRIS は CPF のバックアップを複数作成します。`iris.cpf` ファイルが変更されている場合、InterSystems IRIS は 1 日 1 回、同じディレクトリに `iris.cpf_yyyymmdd` という名前のバックアップを作成します。これらのバックアップは、1 年が経過すると自動的に削除されます。さらに、起動またはシャットダウンが正常に完了した後も、CPF のコピーが、最新の有効な構成を表す `_LastGood.cpf` としてインストール・ディレクトリに保存されます。

1.1.1 CPF の形式

構成パラメータ・ファイルは、行単位の表記の UTF-8 テキスト・ファイルで、拡張子 `.cpf` が付いています。各行の終わりには、キャリッジ・リターンと改行文字を置きます。長い項目を次の行に続けることはできません。ファイル内の 1 行は、以下のいずれかの要素になります。

- ・ [空白スペース](#) – 0 個以上のスペースから成る空の行
- ・ [セクション見出し](#) – 角括弧 ([]) で囲まれたファイル・セクション名
- ・ [パラメータ](#) – 1 個の InterSystems IRIS 構成パラメータとその値
- ・ [コメント](#) – ユーザが追加したコメント

1.1.1.1 空白スペース

一般的に、行の先頭や行の終わりにあるスペースは無視されます。行の途中にあるスペースは、通常は意味があります。行の途中のスペースは、文字列内で意味を持つ要素である場合を除いて、使用しないことをお勧めします。

1.1.1.2 セクション見出し

互いに関連する設定は同じセクションに集められています。セクションの開始を示す行では、セクションの名前が角括弧で囲まれています。以下はその例です。

```
[Devices]
```

セクション見出し以降、次のセクション名（またはファイルの終わり）までの行はすべて、同じセクションに属します。

1.1.1.3 パラメータ

セクション見出しの下にある行はそれぞれ、1 つのパラメータの定義です。パラメータ行の構文は以下のとおりです。keyword はパラメータ名、value は文字列です。

```
keyword=value
```

構成する類似の項目のセットがある場合は、keyword_# と表示される場合があります。例えば、ネームスペース、データベース、デバイスなど、類似の項目または項目のグループを構成する場合は、1 行に 1 項目ずつ指定します。構文は以下のとおりです。

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

value 文字列の構文は、パラメータによって大きく異なります。パラメータに応じて、True または False を表す 1 または 0、バイト数やMB 数などを指定します。また、値が 1 つのみの場合もあれば、複数の値を区切り文字で区切って指定する場合もあります。文字列の中の区切り文字も、コンマ、セミコロン、チルダ (~)、スラッシュ (/)、コロン、およびこれらの組み合わせなど、パラメータによって異なります。

重要 InterSystems IRIS のバージョン 2022.1.1 以降、パラメータ値として空の引用符を設定すると (`parameter=""`)、その値は空の文字列 (値なし) に設定されます (これまでのバージョンでは、このように設定すると既定値に設定されていました)。必須のパラメータ値として空の文字列を設定することはできないので、このようなパラメータを空の引用符に設定するとエラーになります。

1.1.1.4 コメント

CPF ではコメントがサポートされています。コメントは、1 行に記述することも、複数行にわたって記述することもできます。また、行の先頭から始めることも、行の他の内容の後から始めることもできます。

1 行コメントを挿入するには、“;” (セミコロン)、“#” (シャープ記号)、または “//” (2 つのスラッシュ) を使用します。

複数行コメントを挿入するには、“/*” (スラッシュ、アスタリスク) を使用してコメントを開始し、“*/” (アスタリスク、スラッシュ) を使用して終了します。

1.1.2 CPF のサンプル

InterSystems IRIS バージョン 2024.1 の一部として、自動構成された Web サーバと共に Windows システムにインストールした既定の CPF の例を以下に示します。例えば、`[Startup]` セクションの `DefaultPort` パラメータと

WebServerPort パラメータの値として、それぞれ既定のスーパーサーバ・ポート 1972 と既定の Web サーバ・ポート 80 が設定されています。

```
[ConfigFile]
Product=IRIS
Version=2024.1

[Databases]
IRISSYS=C:\InterSystems\IRIS20241\mgr\
IRISLIB=C:\InterSystems\IRIS20241\mgr\irislib\
IRISTEMP=C:\InterSystems\IRIS20241\mgr\iristemp\
IRISLOCALDATA=C:\InterSystems\IRIS20241\mgr\irislocaldata\
IRISAUDIT=C:\InterSystems\IRIS20241\mgr\irisaudit\
ENSLIB=C:\InterSystems\IRIS20241\mgr\enslib\
USER=C:\InterSystems\IRIS20241\mgr\user\

[Namespaces]
%SYS=IRISSYS
USER=USER

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

[Journal]
AlternateDirectory=C:\InterSystems\IRIS20241\mgr\journal\
ArchiveName=
BackupsBeforePurge=2
CompressFiles=1
CurrentDirectory=C:\InterSystems\IRIS20241\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=
ZSTU=1

[WorkQueues]
Default=
SQL=

[Logging]
```

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

[Gateways]
%DotNet Server=.NET,53374,%Gateway_Object,N6.0
%IntegratedML Server=ML,53574,%Gateway_ML
%JDBC Server=JDBC,53774,%Gateway_SQL,,,,,0
%Java Server=Java,53274,%Gateway_Object
%Python Server=Python,53474,%Gateway_Object
%R Server=R,53874,%Gateway_Object
%XSLT Server=XSLT,53674,%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=
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
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=2024-03-07 16:18:42

```

1.2 アクティブな CPF の編集

CPF は、管理ポータル、API 呼び出し、テキスト・エディタなど、複数の方法で操作できます。特定のパラメータを変更する方法の手順については、該当するパラメータのリファレンス・ページの“このパラメータの変更”のセクションを参照してください。変更によっては、変更を有効にするためにインスタンスの再起動が必要になる場合があります。

テキスト・エディタを使用して CPF を変更する場合は、最初にインスタンスをシャットダウンする必要があります。[インストール・ディレクトリ](#)にある `iris.cpf` ファイルを開き、目的の変更を行います。CPF が無効であると InterSystems IRIS が起動しなくなることがあるため、CPF を編集する前にバックアップ・コピーを保存しておくことをお勧めします。必ず、この章の“[CPF の形式](#)”のセクションで説明されている構文に従ってください。

InterSystems IRIS で使用する CPF を `iris start` コマンドで指定したり、部分的な CPF を作成して、UNIX® システムや Linux システムへの導入時に `iris.cpf` にマージしたりできます。これらのオプションは、以下のセクションで説明します。

- ・ [起動時の CPF の選択](#)
- ・ [構成マージ機能の使用](#)

1.2.1 起動時の CPF の選択

開発用、テスト用など、2 つ以上の InterSystems IRIS 構成を頻繁に切り替える場合は、それらの目的に合わせて別個の CPF を作成できます。InterSystems IRIS の起動時に、使用する `.cpf` ファイルを指定すると、手動で設定を変更する時間を減らすことができます。

例えば、Windows で、InterSystems IRIS インストール・ディレクトリが C:\IRIS の場合、各 CPF は以下のようになります。

```
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
```

別の CPF を使用するには、まず InterSystems IRIS を停止する必要があります。続いて、InterSystems IRIS で使用する CPF のフル・パスを指定して、iris start コマンドでインスタンスを起動します。iris start コマンドについては、“システム管理ガイド”の“InterSystems IRIS 複数インスタンスの使用法”の章にある“[InterSystems IRIS インスタンスの制御](#)”のセクションで説明されています。

インスタンスのシャットダウン時に、最後の既知のエラーなし構成が、インストール・ディレクトリ内の _LastGood.cpf というファイルに自動的に保存されます。このファイルは、必要に応じてリカバリに利用できます。

1.3 構成マージ機能の使用

宣言型の構成マージ・ファイルを使用して、既定の iris.cpf を変更できます。マージ・ファイルは、任意の数の CPF パラメータに目的の値を設定する部分的な CPF です。

構成マージはさまざまな目的に役立ちます。マージ・ファイルにより、同じソースから導入する複数のインスタンスの設定を個別に指定し、自動導入や DevOps アプローチをサポートできます。コンテナ化された InterSystems IRIS インスタンスとコンテナ化されていない InterSystems IRIS インスタンスで構成マージを使用できます。詳細は、“構成マージを使用した InterSystems IRIS の自動構成”の“[導入での構成マージ](#)”および“[構成マージを使用した既存のインスタンスの再構成](#)”を参照してください。

構成マージは、自動導入で非常に便利です (UNIX® および Linux システムでのみ)。新規インスタンスの初回起動前に、指定された構成変更を行うため、同じコンテナ・イメージまたは同じインストール・キットから導入する複数のインスタンスの構成をカスタマイズできるからです。マルチノード・トポロジの自動導入では、複数のマージ・ファイルを使用して異なるグループのインスタンスをカスタマイズできます。例えば、[計算ノードを含むシャード・クラスタ](#)の自動導入では、データ・ノード 1、残りのデータ・ノード、および計算ノードに異なるマージ・ファイルをこの順序で適用します。また、[ミラー](#)の導入時には、プライマリ、バックアップ、および非同期メンバに異なるマージ・ファイルを適用できます。

複数のインスタンスの自動再構成も同じ方法で行うことができます。それぞれのインスタンスのグループに適用可能なマージ・ファイルを指定して、インスタンスのグループを再起動します。

[iris merge コマンド](#) (Windows、UNIX、および Linux システムで使用可能) では、構成マージを使用して、実行中のインスタンスを再構成できます。iris merge を使用して、実行している複数のインスタンスに対する同じマージ・ファイルの適用を自動化することによって、これらのインスタンスすべてを同時に同じ方法で再構成でき、アプリケーションまたはクラスタ全体にわたって同じ一連の構成変更を適用できます。前述のとおり、自動化された 1 つのプログラムを使用して、インスタンスのグループごとに (異なるミラー・メンバやクラスタ・ノード・タイプなど) 異なるマージ・ファイルを適用することもできます。

マージ・ファイルの構文は CPF と同じです (“[CPF の形式](#)”を参照)。また、マージ・ファイルには任意の名前と拡張子を付けることができます。マージ・ファイルには CPF 内にある任意のパラメータを記述できますが、変更しない値を含める必要はありません。

CPF とは異なり、マージ・ファイルには同じセクションとパラメータを重複して記述できます。この場合、InterSystems IRIS ではファイルの末尾に近い方の値が優先されるため、これを利用してテンプレート・マージ・ファイルを作成できます。例えば、一般的に必要な値をファイルの先頭に保存し、インスタンス固有の値を末尾に追加した場合、マージ・ファイルの読み取り時にインスタンス固有の値が優先されます。

マージ・ファイルの使用例については、以降のセクションを参照してください。InterSystems IRIS コンテナ導入時のマージ・ファイルの使用法の詳細は、“[構成マージを使用した InterSystems IRIS の自動構成](#)”を参照してください。マージ・

ファイルを InterSystems Kubernetes Operator (IKO) と共に使用方法の詳細は、“InterSystems Kubernetes Operator の使用” の “[configSource: 構成ファイルの作成とその ConfigMap の提供](#)” を参照してください。

1.3.1 構成マージの例

この例では、起動時にマージ・ファイルを使用して、コンテナ化されていないインスタンスの[共有メモリ・ヒープとデータベース・キャッシュ](#)を変更する方法を説明します。これらの設定はそれぞれ `gmheap` パラメータと `globals` パラメータで制御します。

最初のステップはマージ・ファイルを作成することです。以下のサンプル・ファイルには `config_merge.cpf` という名前が付いていますが、任意の名前または拡張子が有効です。マージ・ファイルでは CPF と同じ[構文](#)を使用することに注意してください。

```
# Example configuration merge file.

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

次に、`iris stop` コマンドを使用してマージ対象のターゲット・インスタンスをシャットダウンします。

```
$ sudo iris stop IRIS
```

最後に、以下のスクリプトのように `ISC_CPF_MERGE_FILE` を使用してインスタンスを再起動します。

このスクリプトが実行されると、`iris.cpf` ファイルが `config_merge.cpf` ファイルで指定したとおりに InterSystems IRIS によって変更されます。

```
#!/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
```

インスタンスが起動したらマージは完了です。`iris.cpf` ファイルに `gmheap` 設定と `globals` 設定の目的の値が含まれることを確認します。

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

注釈 `ISC_CPF_MERGE_FILE` で指定されたマージ・ファイルが存在しない場合、インスタンスの起動時にエラー・メッセージが表示されて起動が続行されます。

1.4 構成のセキュリティ

偶発的または意図的な誤った構成から CPF を保護するために、[構成セキュリティ](#)を有効にすることができます。このオプションは、[\[システムワイドセキュリティパラメータ\]](#) ページ ([\[システム管理\]](#)→[\[セキュリティ\]](#)→[\[システム・セキュリティ\]](#)→[\[システムワイドセキュリティパラメータ\]](#)) で利用できます。

[構成セキュリティ](#)が有効になっている場合、InterSystems IRIS の起動時に、前回の起動時以降、構成パラメータ・ファイルが変更されたことが検出され、その変更の検証のために、InterSystems IRIS の起動ユーティリティによってユーザ名とパスワードの入力が要求されます。このユーザ・アカウントには、`%Admin_Manage:Use` 特権が付与されている必要があります。適切なユーザ名とパスワードを指定できない場合、オペレータは以下のいずれかを選択できます。

1. ユーザ名とパスワードを再入力する。
2. 最新の既知の正しい構成を使用して起動する。
3. 起動を中止する。

オペレータが 2 番目のオプションを選択した場合は、起動時に呼び出したパラメータ・ファイルの名前 (file.cpf) が変更され、_rejected という接尾語が付加されます (file.cpf_rejected)。次に、最新の既知の正しい構成 (_LastGood_.cpf) によって file.cpf が上書きされ、この構成を使用して InterSystems IRIS が起動します。

注釈 この**構成セキュリティ**設定は、オペレーティング・システム・レベルのセキュリティの代わりにはなりません。構成ファイルを変更できる機能を、ユーザに対してオペレーティング・システム・レベルで厳格に制限することで、構成ファイルを保護することをお勧めします。

システム全体に適用されるその他のセキュリティ・パラメータの詳細は、“[システム管理およびセキュリティ](#)”を参照してください。

1.5 パラメータの説明

このドキュメントの各パラメータ・リファレンス・ページには、以下のほとんどのセクションが含まれます。

- ・ **構文** – このパラメータが含まれる CPF のセクションと、それに続く構文の概要。その下には、有効な入力の説明と既定値があります。
- ・ **説明** – パラメータの正式な説明。有効な入力の例や、値を選択する際のガイドラインが含まれる場合があります。
- ・ **このパラメータの変更** – このパラメータを変更するためのさまざまな方法。プログラムによる方法、またはブラウザベースの管理ポータルを使用する方法があります。
- ・ **関連項目** – 関連するパラメータと関連ドキュメントへのリンク。

2

[Actions]

[構成マージ機能](#)では、構成パラメータ値を変更できるほか、さまざまな InterSystems IRIS オブジェクトを作成、変更、削除できます。このようなオブジェクトとして、新たに開発したインスタンスにあるか既存のインスタンスにあるかに関係なく、ネームスペースとデータベース、ユーザ、ロール、リソース、ミラーとミラー・メンバなどがあります。これを実行するには、マージ・ファイルでのみ有効な [Actions] セクションにあるパラメータを使用します。

重要 [Actions] セクションを直接構成パラメータ・ファイル (CPF) に追加しないでください。[Actions] セクションは CPF ではサポートされていないため、追加するとインスタンスの起動に失敗します。

[Actions] で指定したオペレーションはベキ等元です。つまり、オペレーションの結果が変化する場合にのみ実行されます。作成するオブジェクトが存在する場合、削除するオブジェクトが存在しない場合、変更するオブジェクトに指定の変更が適用済みの場合、そのオペレーションはスキップされます。構成マージ・ファイルの [Actions] セクションに記述された操作の順序は、操作が実行される順序には影響しません。InterSystems IRIS は確定的な順序で操作を実行します。

構成マージ機能の詳細は、“[構成マージを使用した InterSystems IRIS の自動構成](#)”を参照してください。[Actions] パラメータの使用法、そのパラメータの一覧と詳細な用途はそれぞれ、上記ドキュメントの“[構成マージで、構成以上のカスタマイズは可能か](#)”と“[\[Actions\] パラメータ・リファレンス](#)”を参照してください。

[Archives]

構成パラメータ・ファイルに、[Archives] セクションが含まれる場合があります。これは、[ジャーナル設定の構成](#)で使用するオプションの[ジャーナル・アーカイブ・ターゲット](#)を定義します。

Archives

[ジャーナル設定の構成](#)で使用するジャーナル・アーカイブ・ターゲットを定義します。

構文

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

Name、Type、および Location は文字列です。

説明

構成パラメータ・ファイル (CPF) の `[Archives]` セクションには、このインスタンスに定義されている各ジャーナル・アーカイブ・ターゲットに 1 つのエントリが含まれます。以下を参照してください。

- ・ Name は、アーカイブ・ターゲットの名前です。
- ・ Type は、ターゲットのタイプで、以下のいずれかです。
 - － s3 は、AWS S3 の場所を指定します。
 - － rsync は UNIX® または Windows のディレクトリ指定を使用するオンプレミスの場所を指定します (この場所が Windows ディレクトリの場合、システムではファイルをコピーする際に、実際には rsync ではなく robocopy を使用します)。
- ・ Location は、実際のディレクトリを指定します。形式は Type によって異なります。
 - － s3 の場合、Location の形式は以下のとおりです。

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

以下に例を示します。

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

- － rsync の場合、Location は次のいずれかの形式になります。

- ・ UNIX® の場合：

```
/directoryname/
```

または：

```
server:/directoryname/
```

- ・ Windows の場合：

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

または：

```
drive:\directoryname\
```

詳細は、["ジャーナル・アーカイブ・ターゲットの構成"](#) および ["ジャーナル設定の構成"](#) を参照してください。

このパラメータの変更

管理ポータルの [Archive Targets] ページ ([システム管理]→[構成]→[システム構成]→[Archive Targets]) に移動します。新しいエントリを追加するには、[Create New Archive Target] を選択します。既存のエントリを編集するには、そのエントリの行の [編集] を選択します。

[ComPorts]

このトピックでは、CPF の [ComPorts] セクションのパラメータについて説明します。[ComPorts] セクションは Windows システムにのみ適用されます。

COMn

COM ポートの既定の設定を定義します。Windows システムでのみ有効です。

構文

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

説明

[ComPorts] セクションには、各 COM ポートのエントリが含まれます。これらのエントリによって COM ポートの既定の設定を定義し、ローカル接続経由、またはモデムを接続したシリアルポート経由で InterSystems IRIS® Data Platform にリモート・ログインできるようにします。数値 n は、物理 COM ポート番号を指します。[ComPorts] セクションのエントリ数が多い場合は、n が 2 桁以上になることもあります。

COMn 内のパラメータは Windows プラットフォームにのみ適用されます。

各 COMn エントリには、セミコロンで区切られた 2 つの値があり、これらによって COM ポート番号 n の既定の設定が定義されます。以下のとおりです。

- ・ a – データ・ビット、パリティなどの COM ポート制御パラメータ (バイト位置形式)。バイト位置は、1 から始まります。値の意味は、左から右の順に以下のとおりです。

バイト位置

説明

バイト位置	説明
バイト 1 : モデム制御	<ul style="list-style-type: none"> - '1' モデム制御を使用する (下記の 1801X11 の例で選択されています) - '0' モデム制御を使用しない - ' ' モデム制御の変更なし (既定値)
バイト 2 : データ・ビット	<ul style="list-style-type: none"> - '5' 5 データ・ビット - '6' 6 データ・ビット - '7' 7 データ・ビット - '8' 8 データ・ビット (下記の 1801X11 の例で選択されています) - ' ' ビット・サイズを変更しない (既定値)
バイト 3 : パリティ	<ul style="list-style-type: none"> - '0' パリティなし (下記の 1801X11 の例で選択されています) - '1' 奇数パリティ - '2' 偶数パリティ - '3' マーク・パリティ - '4' スペース・パリティ - ' ' パリティ設定の変更なし (既定値)
バイト 4 : ストップ・ビット	<ul style="list-style-type: none"> - '1' 1 ストップ・ビット (下記の 1801X11 の例で選択されています) - '5' 1.5 ストップ・ビット - '2' 2 ストップ・ビット - ' ' ストップ・ビット設定の変更なし (既定値)
バイト 5 : フロー制御	<ul style="list-style-type: none"> - 'X' Xon/Xoff フロー制御を使用する (下記の 1801X11 の例で選択されています) - 'C' CTS/RTS フロー制御を使用する - 'C' DSR/DTR フロー制御を使用する - ' ' フロー制御の変更なし (既定値)
バイト 6 : DTR 設定	<ul style="list-style-type: none"> - '0' DTR を無効にする。オフに設定、オフを保持。 - '1' DTR を有効にする。オンに設定、オンを保持 (下記の 1801X11 の例で選択されています)。 - ' ' DTR 状態の変更なし (既定値)

バイト位置	説明
バイト 7 : \$ZA エラー報告	<ul style="list-style-type: none"> - '0' \$ZA エラー報告を無効にする - '1' \$ZA エラー報告を有効にする (下記の 1801X11 の例で選択されています) - ' ' \$ZA エラー報告の変更なし (既定値)

- ・ b - ボー・レート。指定しない場合、既定のボー・レートは 19200 です。

例

以下の [ComPorts] セクションの例は、COM ポート制御パラメータで値としてスペース文字を使用する方法を示しています。

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

最初の例では、COM ポート制御パラメータに対してすべて既定値を使用するために、セミコロン区切り文字の前に 7 個のスペースを指定しています。この意味は、「モデム制御の変更なし、ビット・サイズの変更なし、パリティ設定の変更なし、ストップ・ビット設定の変更なし、フロー制御の変更なし、DTR 状態の変更なし、\$ZA エラー報告の変更なし」です。

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

2 番目の例では、COM ポート制御パラメータの値として 1801X11 を指定しています。この意味は、「モデム制御を使用する、8 データ・ビット、パリティなし、1 ストップ・ビット、Xon/Xoff フロー制御を使用する、DTR を有効にする、\$ZA エラー報告を有効にする」です。

このパラメータの変更

テキスト・エディタで CPF を編集してこのパラメータを変更できます ("[アクティブな CPF の編集](#)" を参照)。

[config]

このトピックでは、CPF の [config] セクションの **[メモリ詳細設定]** のパラメータについて説明します。

LibPath

LD_LIBRARY_PATH 環境変数にディレクトリを追加します。UNIX® システムでのみ有効です。

構文

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

directory は、有効なディレクトリのフル・パスです。LibPath の最大長は 1024 文字です。既定では、ディレクトリは記述されていません。

説明

LibPath は UNIX® システムでのみ使用されます。このパラメータによって、サードパーティ製共有ライブラリの検索のために、ディレクトリのリストが LD_LIBRARY_PATH 環境変数 (macOS では DYLD_LIBRARY_PATH) に追加されます。この設定を変更した場合は、インスタンスを再起動して、変更を有効にする必要があります。

macOS では、システム整合性保護 (SIP) が有効になっている場合、システム・ディレクトリ内のプログラムを実行する DYLD_LIBRARY_PATH 変数が無視される可能性があります。

このパラメータの変更

管理ポータル の **[メモリ詳細]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[メモリ詳細]**) で、**[LibPath]** の行の **[編集]** を選択します。1 つ以上のディレクトリをコロンで区切って入力します。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、LibPath を変更できます。

MaxServerConn

ECP クライアントからの接続の最大数を設定します。

構文

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

n は、0 から 254 の範囲の整数です。既定値は 1 です。

説明

MaxServerConn は、このインスタンスに同時にアクセス可能な ECP クライアントの最大数です。これは、このインスタンスが ECP サーバとして稼動しているときに受け入れることができる接続の最大数です。この設定を変更した場合は、インスタンスを再起動して、変更を有効にする必要があります。

インスタンスが[シャード・クラスタ](#)のメンバである場合、この設定はクラスタ内のノードの数以上にする必要があります。

このパラメータの変更

管理ポータル の [\[メモリ詳細\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[メモリ詳細\]](#)) で、[\[MaxServerConn\]](#) の行の [\[編集\]](#) を選択します。値を入力します。

このパラメータは、管理ポータル の [\[ECP設定\]](#) ページで変更することもできます ([\[システム管理\]](#) > [\[構成\]](#) > [\[接続性\]](#) > [\[ECP設定\]](#))。 [\[このシステムを ECP データサーバとする\]](#) の列で、[\[アプリケーションサーバの最大数\]](#) を編集します。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (["アクティブな CPF の編集"](#) を参照)、MaxServerConn を変更できます。

MaxServers

ECP サーバへの接続の最大数を設定します。

構文

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

n は、0 から 254 の範囲の整数です。既定値は 2 です。

説明

MaxServers は、このインスタンスからアクセス可能な ECP サーバの最大数です。これは、このインスタンスが ECP クライアントとして稼動しているときに確立できる接続の最大数です。この設定を変更した場合は、インスタンスを再起動して、変更を有効にする必要があります。

インスタンスが[シャード・クラスタ](#)のメンバである場合、この設定はクラスタ内のノードの数以上にする必要があります。

このパラメータの変更

[システム管理] > [構成] > [接続性] > [ECP 設定] ページで、[このシステムを ECP アプリケーションサーバとする] 列の [データサーバの最大数] を編集します。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、MaxServers を変更できます。

Path

既定の PATH 環境変数にディレクトリを追加します。UNIX® システムでのみ有効です。

構文

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

directory は、有効なディレクトリのフル・パスです。Path の最大長は 1024 文字です。既定では、ディレクトリは記述されていません。

説明

InterSystems IRIS® データ・プラットフォームの起動の一環として、UNIX® システムでは、新しいプロセスに UNIX® PATH 環境変数を割り当てます。既定では、UNIX® PATH 変数は以下のとおりです。

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

PATH を使用する InterSystems IRIS プロセスとしては、システム・デーモン、SYSTEM^%ZSTART によって開始されるプロセス、スーパー・サーバによって作成されるプロセス (JDBC/ODBC サーバなど) などがあります。

お客様のアプリケーションにおいて、InterSystems IRIS が提供する既定の UNIX® PATH に追加された検索ディレクトリをこれらのプロセスの PATH 環境変数に含める必要がある場合があります。CPF path 変数を使用することで、このパスにディレクトリを追加できます。この設定を変更した場合は、インスタンスを再起動して、変更を有効にする必要があります。

注釈 ターミナル・プロセスではそれらの PATH がこのように設定されないため、ログイン・スクリプトでそれらの PATH を設定する必要があります。

例

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

これによって、システム・プロセスの PATH 環境変数が以下に設定されます。

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

このパラメータの変更

管理ポータル の **[メモリ詳細]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[メモリ詳細]**) で、**[Path]** の行の **[編集]** を選択します。1 つ以上のディレクトリをコロンで区切って入力します。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("**アクティブな CPF の編集**" を参照)、Path を変更できます。

PythonPath

組み込み Python の `sys.path` に追加するディレクトリを指定します。

構文

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

directory は、有効なディレクトリへの 1 つ以上のパスです。PythonPath の最大長は 1024 文字です。既定では、ディレクトリは記述されていません。

説明

モジュールをインポートする際に、組み込み Python では `sys.path` 変数を使用して、Python パッケージを配置できる場所を指定します。PythonPath は、Python パッケージをインストールする予定の追加ディレクトリを指定し、モジュールをインポートする際に組み込み Python がそれらのディレクトリを特定できるようにします。

複数のディレクトリを指定できます。Windows ではセミコロンを使用して複数のディレクトリを区切り、他のプラットフォームではコロンを使用して複数のディレクトリを区切ります。

例

Ubuntu では、`PythonPath=/home/user/mypackages:/home/user/morepackages` によって、`sys.path` に 2 つのディレクトリが追加されます。

ターミナルからの以下の例は、組み込み Python の起動時に、指定したディレクトリが `sys.path` に存在することを示しています。

```
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', '/home/user/mypackages',
'/home/user/morepackages', '/bin/local/lib/python3.10/dist-packages', '/bin/lib/python3/dist-packages',
'/bin/lib/python3.10/dist-packages']
```

このパラメータの変更

管理ポータル の [\[メモリ詳細\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[メモリ詳細\]](#)) で、[\[PythonPath\]](#) の行の [\[編集\]](#) を選択します。1 つ以上のディレクトリを入力します。

管理ポータルを使用する代わりに、`Config.config` クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、PythonPath を変更できます。

この設定を編集すると、即座に反映されます。新しいパスは、新しいプロセスすべてで使用されます。

PythonRuntimeLibrary

組み込み Python を実行する際に使用するランタイム・ライブラリの場所を指定します。

構文

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

library は、Python ランタイム・ライブラリの絶対パスとファイル名です。PythonRuntimeLibrary の最大長は 1024 文字です。既定では、ライブラリは記述されていません。

説明

PythonRuntimeLibrary は、組み込み Python を実行する際に使用する Python ランタイム・ライブラリの場所を指定します。このバージョンのライブラリは、組み込み Python の起動時に使用される既定のバージョンのライブラリをオーバーライドします。

例：PythonRuntimeLibrary=/usr/lib/x86_64-linux-gnu/libpython3.11.so.1

注釈 この機能の詳細は、“[フレキシブル Python ランタイム機能](#)”を参照してください。

この機能を使用できるプラットフォームについては、“[その他のサポート対象機能](#)”を参照してください。

このパラメータの変更

管理ポータル[の](#) [\[メモリ詳細\]](#) ページ ([\[システム管理\]](#)→[\[構成\]](#)→[\[追加の設定\]](#)→[\[メモリ詳細\]](#)) で、[\[PythonRuntimeLibrary\]](#) の行の [\[編集\]](#) を選択します。使用する Python ランタイム・ライブラリの場所を入力します。

管理ポータルを使用する代わりに、Config.config クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)”を参照)、PythonRuntimeLibrary を変更できます。

この設定を編集すると、即座に反映されます。新しいランタイム・ライブラリは、新しいプロセスすべてで使用されます。

bbsiz

プロセスあたりの最大メモリを設定します。

構文

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

n は、256 から 2,147,483,647 (KB) の範囲の整数です。既定値は -1 です。パラメータが最大値 (2,147,483,647 KB) に設定されます。

説明

bbsiz は、1 プロセスへの割り当てを許可するメモリの最大量 (KB) です。このプロセス・プライベート・メモリの容量は、シンボル・テーブル割り当て、および I/O デバイスなどからのさまざまなメモリ要求に使用します。割り当て容量は、アプリケーションにおける必要性に応じて最大値に達するまで増加します。

一度メモリがプロセスに割り当てられると、通常はプロセスが終了するまで、その割り当ては解除されません。ただし、大量の (例えば 32MB を超える) メモリが使用されてから解放された場合、可能であれば、InterSystems IRIS® Data Platform は割り当て解除されたメモリをオペレーティング・システムに戻します。

このパラメータの変更

管理ポータルの [\[メモリと開始設定\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[システム構成\]](#) > [\[メモリと開始設定\]](#)) で、[\[プロセスあたりの最大メモリ \(KB\)\]](#) 行に KB 単位の数値を入力します。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、bbsiz を変更できます。

この設定を編集した場合、更新された値は新しいプロセスすべてに適用されます。

既定値の -1 (最大値である約 2 テラバイト) に設定すると、プロセス・メモリあたりの最大サイズは実質的に無制限になります。プロセスがそれほど多くのメモリを使用する状態になる可能性はきわめて低いからです。

console

メッセージ・ログ・ファイル (messages.log) のパスを設定します。

構文

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

ConsoleFile は、**messages.log** ファイルのフル・パスです。ConsoleFile の最大長は 227 文字です。既定では、ディレクトリは記述されていません。

説明

console には、**messages.log** ファイルを構成する以下の 2 つのコンマ区切り値が含まれます。

ConsoleFile

messages.log ファイルのパス。InterSystems IRIS® データ・プラットフォームはここにメッセージを記録します。ConsoleFile を変更して、ログ・ファイルの新しい場所を指定します。ただし、ログ・ファイルの名前は常に **messages.log** でなければなりません。値を指定しない場合は、install-dir/mgr/messages.log という名前のファイルに出力されます。

ConsoleFile は、**console** パラメータの 2 つ目のコンマ区切り値です。

メッセージ・ログは、管理ポータルの **[メッセージログ]** ページ (**[システム処理]** > **[システムログ]** > **[メッセージログ]**) で参照できます。**messages.log** ファイルの詳細を構成するには、“**MaxConsoleLogSize**” パラメータを参照してください。

VMSConsoleTerminal

未使用。

このパラメータの変更

管理ポータルの **[メモリ詳細]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[メモリ詳細]**) で、**[ConsoleFile]** の行の **[編集]** を選択します。ディレクトリ・パスを入力します。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“**アクティブな CPF の編集**” を参照)、ConsoleFile を変更できます。

errlog

エラー・ログの最大エントリ数を設定します。

構文

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

n は、10 から 10,000 の範囲の整数です。既定値は 500 です。

説明

errlog は、InterSystems IRIS® データ・プラットフォーム・システム・エラー・ログの最大項目数です (詳細は、"[InterSystems IRIS システム・エラー・ログ](#)" を参照してください)。この上限に達すると、ログ・ファイルの古いエントリは無効になります。

このパラメータの変更

管理ポータル の **[メモリ詳細]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[メモリ詳細]**) で、**[errlog]** の行の **[編集]** を選択します。エントリの数を入力します。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、errlog を変更できます。

globals

データベース・キャッシュに共有メモリを割り当てます。

構文

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

説明

globals には、各ブロック・サイズに対してデータベース・キャッシュに割り当てる共有メモリの量を指定する 6 つのコンマ区切り値が含まれます。左から右の順に、各値は、以下に対して割り当てる MB 単位の数値を表します。

- ・ a - 現在は使用されていません。常に 0 です。
- ・ b - 現在は使用されていません。常に 0 です。
- ・ c - 8 KB ブロック
- ・ d - 16 KB ブロック
- ・ e - 32 KB ブロック
- ・ f - 64 KB ブロック

6 つの値をすべて 0 (既定値) に設定した場合、自動的に合計物理メモリの 25% が割り当てられます。64 ビット・システムでは、16 TB の上限があります。

このパラメータの変更

既定の 8 KB ブロック以外のブロック・サイズでデータベースを作成するには、[Startup] セクションの [DBSizesAllowed](#) パラメータを使用して追加のブロック・サイズを有効にする必要があります。

データベース・キャッシュへのメモリの割り当て (管理ポータルを使用した割り当てなど) の詳細は、“[データベース・キャッシュおよびブルーチン・キャッシュへのメモリの割り当て](#)” を参照してください。

Config.config クラスを使用するか、[CPF を編集することによって](#) を変更することもできます。

関連項目

- ・ [routines](#) パラメータ
- ・ このリファレンスの [Startup] セクションの [DBSizesAllowed](#) パラメータ
- ・ [メモリ要件の見積もり](#)。
- ・ [データベース・キャッシュおよびブルーチン・キャッシュへのメモリの割り当ておよびラージ・ブロック・サイズに関する考慮事項](#)

gmheap

共有メモリ・ヒープのサイズを設定します。

構文

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

n は、2048 から 1,073,741,760 (KB) の範囲の整数です。既定値は 0 です。システムのサイズに基づいて、共有メモリ・ヒープのサイズが合理的な値に設定されます。

説明

gmheap は、InterSystems IRIS® データ・プラットフォームの共有メモリ・ヒープ (旧名は一般メモリ・ヒープ) のサイズ (KB) です。共有メモリは、グローバル・マッピング、データベース名とディレクトリの情報、セキュリティ・システムなど特定の目的用に必要に応じてこの合計から割り当てられます。所定の時間に所定のサブシステムで使用されている共有メモリは、現在割り当てられているメモリより小さい場合があります。

共有メモリの割り当ては [\[共有メモリヒープ使用状況\]](#) ページに表示されます ([\[システムオペレーション\]](#) → [\[システム使用\]](#) ページに移動し、[\[共有メモリヒープ使用状況\]](#) ボタンをクリックします)。詳細は、["共有メモリ・ヒープ使用状況"](#) を参照してください。このページはメモリ割り当ておよび使用量をバイト単位で表示しますが、共有メモリはページ単位で割り当てられます。

既定では、gmheap は 0 に設定されています。これにより、全体のシステム・サイズに基づいて合理的なサイズが自動的に選択されます。0 を選択した場合、gmheap のサイズは、[グローバル・バッファ](#) 向けに構成した合計メモリの 3% になるように構成されます。gmheap の最小値は 307,200 KB (300 MB) に構成され、最大値は 2,097,000 KB (2 GB) に構成されます。これよりも大きいメモリ量や小さいメモリ量を手動で構成することもできます。

場合によっては、使用可能な共有メモリを十分に確保するために gmheap を増やす必要があります。その例を以下に示します。

- ・ ジャーナル・ファイルのリストア

ジャーナル・リストアで最適なパフォーマンスを実現するには、共有メモリ・ヒープのサイズを大きくすることをお勧めします。詳細は、["並列デジャーナリングのシステム要件"](#) を参照してください。

- ・ SQL クエリを並列実行しているとき

クエリの並列実行では共有メモリ・ヒープから共有メモリが余分に使用されるので、並列クエリで最適なパフォーマンスを実現するには gmheap を大きくすることが必要なこともあります。詳細は、["共有メモリの考慮事項"](#) を参照してください。

locksiz 設定では、使用可能な合計共有メモリのうちロックの管理に特別に割り当てることができる部分が構成されます (ロック・テーブル)。locksiz は gmheap のサブセットで、gmheap の残りの部分はその他のサブシステムすべてに使用できます。よって、この関係を考慮して gmheap と locksiz のサイズを決定し、locksiz を増やしたら gmheap もそれに比例して増やすことが重要です。

このパラメータの変更

管理ポータル [\[メモリ詳細\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[メモリ詳細\]](#)) で、[\[gmheap\]](#) の行の [\[編集\]](#) を選択します。KB 単位の数値を入力します。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (["アクティブな CPF の編集"](#) を参照)、gmheap を変更できます。

gmheap 設定を編集した場合、変更内容を適用するには、InterSystems IRIS を再起動する必要があります。

history

コマンド行呼び出しオプションを定義します。

構文

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

n は、0 から 1000 の範囲の整数です。既定値は 500 です。

説明

history は、コマンド行/行呼び出しバッファに保持されるエントリの最大数です。

このパラメータの変更

管理ポータルの [\[メモリ詳細\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[メモリ詳細\]](#)) で、[\[history\]](#) の値を編集します。

管理ポータルを使用する代わりに、Config.config クラス ([クラス・リファレンスを参照](#)) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、history を変更できます。

この設定を適用するために再起動する必要はありません。

ijcbuf

インタジョブ・コミュニケーション・バッファのサイズを設定します。

構文

```
[config]    ijcbuf=n
```

n は、512 から 8192 (バイト) の範囲の整数です。既定値の 512 を使用することをお勧めします。

説明

ijcbuf は、インタジョブ・コミュニケーション (IJC) バッファごとに割り当てるバイト数です。詳細は、“[ローカル・プロセス間通信 \(パイプ\)](#)” を参照してください。パラメータ [ijcnum](#) も参照してください。

このパラメータの変更

管理ポータル の [\[メモリ詳細\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[メモリ詳細\]](#)) で、[\[ijcbuf\]](#) の行の [\[編集\]](#) を選択します。バイト単位の数値を入力します。

管理ポータルを使用する代わりに、Config.config クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、ijcbuf を変更できます。

この設定を編集した場合、変更内容を適用するには、InterSystems IRIS® Data Platform を再起動する必要があります。

ijcnum

インタジョブ・コミュニケーション・デバイスの数を設定します。

構文

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

n は、0 から 256 の範囲の整数です。既定値の 16 デバイスを使用することをお勧めします。

説明

ijcnum は、インタジョブ・コミュニケーション (IJC) デバイスの数です。各デバイスは、“[ijcbuf](#)” で定義されている IJC バッファ・サイズに対応します。詳細は、“[ローカル・プロセス間通信 \(パイプ\)](#)” を参照してください。

このパラメータの変更

管理ポータル の [\[メモリ詳細\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[メモリ詳細\]](#)) で、[\[ijcnum\]](#) の行の [\[編集\]](#) を選択します。デバイスの数を入力します。

管理ポータルを使用する代わりに、Config.config クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、ijcnum を変更できます。

この設定を編集した場合、変更内容を適用するには、InterSystems IRIS® Data Platform を再起動する必要があります。

jrnbufs

ジャーナル・バッファ用のメモリを割り当てます。

構文

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

n は整数です。最大値は 1024 (MB) です。最小値は、Unicode インスタンスの場合は 16、8 ビット・インスタンスの場合は 8 です。既定値は 64 です。

説明

jrnbufs は、ジャーナル・バッファに割り当てられるメモリの容量です。この設定値を増やすと、メモリに保管できるジャーナル・データの容量が増大するため、ジャーナリングのパフォーマンスが向上しますが、システム障害時に失われる可能性があるジャーナル・データの容量も増大します。

jrnbufs を増やすと、ジャーナル・ファイルのサイズ ([管理ポータル](#)で、`^JRNOPTS` を使用して、または `FileSizeLimit` パラメータを設定して構成) が縮小する可能性があります。これは、2 つの設定の合計サイズが 4 GB に制限されているためです。詳細は、“[ジャーナル設定の構成](#)” を参照してください。

このパラメータの変更

管理ポータルの [\[メモリ詳細\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[メモリ詳細\]](#)) で、`[jrnbufs]` の行の [\[編集\]](#) を選択します。MB 単位の数値を入力します。

管理ポータルを使用する代わりに、`Config.config` クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、jrnbufs を変更できます。

この設定を編集した場合、変更内容を適用するには、InterSystems IRIS® Data Platform を再起動する必要があります。

locksiz

ロックに使用する共有メモリの最大サイズを設定します。

構文

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

n は整数で、最小値は 65,536 (バイト) です。既定値は 0 です。すべてのシステムで適切になるように設計された値に設定されます。locksiz を 0 に設定すると、gmheap パラメータで指定された共有メモリ・サイズのみでこのサイズが制限されます。

説明

既定では、すべてのシステムで適切になるように設計された値に locksiz が構成されます。ただし、ロックの割り当てに使用するメモリは gmheap (共有メモリ・ヒープ) から取得することから、gmheap に存在するよりも多くのメモリをロックに使用することはできません。ロック・テーブルにそれ以上の領域が必要な場合は、gmheap パラメータを大きくしてください。LockThreshold を低くして、ロック・テーブル内の領域の使用量を少なくすることもできます。

この設定を編集した場合、変更は即座に反映されます。

このパラメータの変更

管理ポータル の **[メモリ詳細]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[メモリ詳細]**) で、[locksiz] の行の **[編集]** を選択します。バイト単位の数値を入力します。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("**アクティブな CPF の編集**" を参照)、locksiz を変更できます。

関連項目

- ・ [ロックと並行処理の制御](#)
- ・ [gmheap](#) パラメータ
- ・ [LockThreshold](#) パラメータ

memlock

共有メモリまたはテキスト・セグメントをメモリ内でロックするよう指定します。

構文

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

n は、ビット・フラグのセットです。既定では、n は 0 です (すべてのフラグが false に設定されています)。

説明

memlock は、InterSystems IRIS® Data Platform が共有メモリを割り当てる方法を制御するビット・フラグのセットです。InterSystems IRIS は、制御構造、グローバル・バッファ、ルーチン・バッファ、および共有メモリ・ヒープで使用する共有メモリ・セグメントを起動時に割り当てます。memlock パラメータを使用することで、その割り当て方法を詳細に制御できます。

既定 (n = 0) では、InterSystems IRIS は、ラージ・ページをサポートするプラットフォーム (Windows、Linux、および AIX) では、以下のようにラージ・ページから共有メモリを割り当てようとします。

1. 可能な場合は、ラージ・ページを要求します。ラージ・ページは、オペレーティング・システム・レベルで物理メモリに自動でロックされます。
2. 構成されているすべてのメモリ容量をラージ・ページで割り当てることができない場合、標準のページ (スモール・ページ) を要求します。標準のページは物理メモリにロックされません。
3. 構成されているすべてのメモリ容量をスモール・ページで割り当てることができない場合は、割り当て量を 1/8 減らして、手順 1 から再実行します。

以下のビット・フラグは、下記のようにこのプロセスを変更します。

1 (LockSharedMemory)

この memlock フラグは、ラージ・ページが使用されていない場合に共有メモリを物理メモリにロックするかどうかを指定します。既定では、ロックしません。これは、Microsoft Windows と macOS を除くすべてのオペレーティング・システムに適用されます。

8 (LockTextSegment)

この memlock フラグは、テキスト・セグメント (InterSystems IRIS の実行可能コード・スペース) を物理メモリにロックするかどうかを指定します (一部の UNIX プラットフォームが対象)。既定では、ロックしません。

32 (LargePagesDisabled)

この memlock フラグは、ラージ/ヒュージ・ページをサポートするプラットフォームで、共有メモリに対してラージ/ヒュージ・ページを無効にするかどうかを指定します。既定では、ラージ/ヒュージ・ページは使用されません。

ラージ・ページをサポートするプラットフォームでこのフラグがオフになっている場合、InterSystems IRIS はラージ・ページでメモリを割り当てようと試み、要求されたサイズでラージ・ページを割り当てることができない場合は標準のページに切り替えます。専門的に言えば、これは、InterSystems IRIS がページ・サイズに対して中立的な処置をとり、ラージ・ページを要求するアクションを実行しないことを意味します。

64 (LargePagesRequired)

この memlock フラグは、ラージ/ヒュージ・ページをサポートするプラットフォーム (Windows、AIX、および Linux) で、共有メモリに対してラージ/ヒュージ・ページの使用を必須にするかどうかを指定します。既定では、必須ではありません。このフラグは、その他のプラットフォームの場合、または LargePageDisabled フラグによってラージ・ページが無効になっている場合は無視されます。

`LargePagesRequired` が `True` で (かつ無視されていない状態で)、ラージ/ヒュージ・ページでメモリを割り当てることができない場合、スモール・ページを代わりに使用せずに起動が中止されます。InterSystems IRIS はメモリ・サイズを少し減らして再試行しますが、このフラグが存在しないときほどは減らしません。

128 (BackoffDisabled)

この `memlock` は、メモリの割り当てに失敗した場合に、容量を減らして再試行するかどうかを指定します。既定では、再試行しません。このフラグが `True` の場合に、構成されたサイズでメモリを割り当てることができないときは、起動は中止されます。

このパラメータの変更

管理ポータル の `[メモリ詳細]` ページ (`[システム管理]` > `[構成]` > `[追加設定]` > `[メモリ詳細]`) で、`[BackoffDisabled]`、`[LargePagesDisabled]`、`[LargePagesRequired]`、`[LockSharedMemory]`、および `[LockTextSegment]` で `True` または `False` を選択します。

管理ポータルを使用する代わりに、`Config.config` クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、`memlock` を変更できます。

この設定を編集した場合、変更内容を適用するには、InterSystems IRIS を再起動する必要があります。

netjob

リモート・ジョブ要求を許可します。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

netjob が有効な場合 (n = 1)、ECP を介して着信するリモート・ジョブ要求は、このサーバで処理されます。

このパラメータの変更

管理ポータルの **[メモリ詳細]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[メモリ詳細]**) で、[netjob] の行の **[編集]** を選択します。[はい] または [いいえ] を選択します。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、netjob を変更できます。

nlstab

NLS 照合テーブルの数を設定します。

構文

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

n は、0 から 64 の範囲の整数です。既定値は 50 です。

説明

nlstab は、InterSystems IRIS® Data Platform の起動時に割り当てる NLS 照合テーブルの数です。このパラメータは、ロード可能な国固有照合テーブルの数を指定するもので、組み込み照合の数は除外されています。

このパラメータの変更

管理ポータル の **[メモリ詳細]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[メモリ詳細]**) で、**[nlstab]** の行の **[編集]** を選択します。テーブルの数を入力します。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、nlstab を変更できます。

overview

インスタンスのプラットフォームとバージョンの概要を確認します。

構文

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

a および b は読み取り専用文字列です。

説明

overview には、プラットフォームおよびバージョンの情報を示す、チルダ (^) で区切られた 2 つの値が表示されます。値の意味は、左から右の順に以下のとおりです。

- ・ a : InterSystems IRIS® データ・プラットフォームを実行する特定のオペレーティング・システム
- ・ b : オペレーティング・システムの一般的な種類

例

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

```
overview=Linux (Intel)~UNIX®
```

pijdir

未使用。

説明

未使用。

routines

ルーチン・キャッシュに共有メモリを割り当てます。

構文

```
[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.
```

このパラメータの既定値は `routines=0` です。この設定では、`globals` パラメータによってデータベース・キャッシュで 8 KB のグローバル・バッファに割り当てられているメモリの 10% に等しいメモリが割り当てられます (最小 80 MB、最大 1 GB)。

説明

`routines` パラメータは、ルーチンのバッファリング用に割り当てるメモリの量を指定します。メモリの合計量は、バッファ・サイズが異なる (2 キロバイトから 64 キロバイト) 6 つのプールで共有されます。

ルーチン・バッファ・プールは 2 つの方法のいずれかで構成できます。その方法とは、**単一値**を指定して自動割り当てる方法か、**複数值**を指定してバッファを手動で割り当てる方法です。一般的なプロダクション・インスタンスの場合は、自動割り当てで十分です。ただし、指定されたアプリケーションの理想的な割り当てはさまざまな要素によって異なるため、パフォーマンスを最適化するには調整が必要となる場合があります。

自動 (単一値) バッファ割り当て

InterSystems IRIS® データ・プラットフォームがルーチン・バッファに割り当てるメガバイトの合計数として、単一値 `n` を指定できます。最小のサイズは 80 MB です。これより小さい値を指定すると、インスタンスによって最大で 80 MB に調整されます。値 0 (既定値) を指定すると、`globals` パラメータによってデータベース・キャッシュで 8 KB のグローバル・バッファに割り当てられているメモリの 10% に等しいメモリが割り当てられます (最小 80 MB、最大 1020 MB)。

InterSystems IRIS によって、このメモリが 4 KB、16 KB、64 KB の各バッファ・プール間で以下のように分割されます。

- ・ 4 KB バッファに 12.5%
- ・ 16 KB バッファに 37.5%
- ・ 64 KB バッファに 50%

例えば、`routines=500` を指定した場合、InterSystems IRIS は以下を作成します。

- ・ 16,000 個の 4 KB バッファ (62.5 MB)
- ・ 12,000 個の 16 KB バッファ (187.5 MB)
- ・ 4,000 個の 64 KB バッファ (250 MB)

手動 (複数值) バッファ割り当て

ルーチン・バッファ・プールごとに割り当てるメモリの量を指定することができます。その場合は、6 つすべての値を指定する必要があります。6 つに満たない場合、InterSystems IRIS は、指定された最初の値を使用して上述の自動形式に戻ります。6 つの値は以下を表します。

- ・ `n1` は 2 KB のルーチン・バッファに割り当てる MB 値です。
- ・ `n2` は 4 KB のルーチン・バッファに割り当てる MB 値です。
- ・ `n3` は 8 KB のルーチン・バッファに割り当てる MB 値です。
- ・ `n4` は 16 KB のルーチン・バッファに割り当てる MB 値です。
- ・ `n5` は 32 KB のルーチン・バッファに割り当てる MB 値です。

- ・ n6 は 64 KB のルーチン・バッファに割り当てる MB 値です。インスタンスは n6 の値に関係なく、常に少なくとも 430 の 64 KB のルーチン・バッファを割り当てます。

例えば、`routines=0,128,128,0,0,800` を指定した場合は以下のように作成されます。

- ・ 128 MB に 4 KB バッファ
- ・ 128 MB に 8 KB バッファ
- ・ 800 MB に 64 KB バッファ

64 KB を除き、特定のサイズのバッファを 0 個にすることはできますが、0 よりも大きい次の最小値は 430 になります。これより小さい数を指定しても、インスタンスによって 430 個のバッファが割り当てられます。バッファの合計最大数は 33,554,432 です。InterSystems IRIS ルーチンのフォーマットでは、最大ルーチン・サイズの設定にかかわらず、リテラル文字列に 32,768 文字を超える文字数を使用することはできません。

このパラメータの変更

ルーチン・キャッシュへのメモリの割り当て (管理ポータルを使用した割り当てなど) の詳細は、“[データベース・キャッシュおよびルーチン・キャッシュへのメモリの割り当て](#)” を参照してください。

`Config.config` クラスを使用するか、[CPF を編集することによって](#)、`routines` を変更することもできます。

関連項目

- ・ [globals](#) パラメータ
- ・ [メモリ要件の見積もり](#)
- ・ [データベース・キャッシュおよびルーチン・キャッシュへのメモリの割り当て](#)

udevtabsize

デバイス・テーブルの最大サイズを設定します。

構文

```
[config]    udevtabsize=n
```

n は、0 から 65535 (バイト) の範囲の整数です。既定値は 24,576 です。

説明

udevtabsize は、デバイス・テーブルの最大サイズ (バイト) です。このテーブルは、デバイス番号 (従来の論理ユニット番号) をデバイス名にマッピングして、ObjectScript コードが番号でデバイスを開くことができますようにします。

このパラメータの変更

[システム管理] > [構成] > [追加設定] > [メモリ詳細] ページで、[udevtabsize] の行の **[編集]** を選択します。バイト単位の数値を入力します。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、udevtabsize を変更できます。

wjdir

ライト・イメージ・ジャーナル・ファイル・ディレクトリを設定します。

構文

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

n は、有効なディレクトリのフル・パスです。最大長は 226 文字です。既定では、ディレクトリは記述されていません。

説明

wjdir は、ライト・イメージ・ジャーナル・ファイルを格納するディレクトリの名前です。ジャーナル・ディレクトリは、データベースとは異なるパーティションに配置することをお勧めします。

値を指定しない場合、InterSystems IRIS® Data Platform は install-dir/**mgr** ディレクトリを使用します。

このパラメータの変更

管理ポータル の [ジャーナル設定] ページ ([システム管理] > [構成] > [システム構成] > [ジャーナル設定]) で、[ライトイメージジャーナルディレクトリ] の行の [参照] を選択します。ディレクトリ名を選択します。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、wjdir を変更できます。

targwjsz

WIJ の目的のサイズを設定します。

構文

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

n は整数です。既定値は 0 です。

説明

targwjsz は、ライト・イメージ・ジャーナル (WIJ) ファイルに MB 数を割り当てます。値は整数ですが、小数を入力することもでき、小数は通知なしで切り捨てられます (例えば、35.5 は 35 になります)。

targwjsz を 0 に設定すると、WIJ は、アクティビティに基づいて必要に応じて拡張されます。targwjsz パラメータを使用すると、WIJ に事前に領域を割り当てることができるため、高アクティビティ期間中に拡張する必要がありません。targwjsz に上限はありませんが、[データベース・キャッシュ](#)の最大サイズよりも大きい値に設定することは有用ではありません。これが、WIJ が必要とする最大領域であるためです。

詳細は、“[ライト・イメージ・ジャーナリングとリカバリ](#)” を参照してください。

注釈 このターゲット・サイズを設定することで、起動プロセスの早い段階で WIJ へのディスク容量割り当てを確保できます。早い段階で十分な容量が割り当てられず、WIJ 用に十分な利用可能領域がないと、インスタンスに問題が発生する可能性があります。WIJ への領域の割り当ては、高度な構成設定です。これに関する問題が発生した場合は、インターシステムズの[サポート窓口](#)までお問い合わせください。

このパラメータの変更

管理ポータルの [\[メモリ詳細設定\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[メモリ詳細\]](#)) で、[\[targwjsz\]](#) の行の [\[編集\]](#) を選択します。[\[編集:targwjsz\]](#) ページに設定の詳細が表示され、このページでその値を変更できます。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、targwjsz を変更できます。

zfheap

\$ZF ヒープのサイズと構成を定義します。

構文

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

ZFString は、0 から 32,767 の範囲の整数です。既定値は 0 です。

ZFSize は、0 から 270,336 (バイト) の範囲の整数です。既定値は 0 です。

説明

zfheap には、\$ZF ヒープを構成する 2 つのコンマ区切り値 ZFString および ZFSize が含まれます。\$ZF ヒープの詳細は、“[コールアウト・ライブラリの作成](#)”を参照してください。

ZFString

この zfheap パラメータは、InterSystems IRIS® Data Platform において \$ZF ヒープで 1 つの文字列パラメータに対して使用できる文字の数です。このために実際に必要なバイト数は、使用している文字が Unicode (2 バイト文字) か、UNIX® 上の 1 バイト文字かに応じて異なります。0 に設定すると、自動的に ZFString のシステム既定値 (32,767) が使用されます。

ZFSize

この zfheap パラメータは、InterSystems IRIS® Data Platform があらゆる目的で \$ZF ヒープに割り当てるバイト列のバイト数です。\$ZF ヒープは、\$ZF のすべての入力および出力パラメータ用に仮想メモリに割り当てられるバイトの合計バイト数です。最初の値で使用可能にした文字列用の領域もこの中に含まれます。0 に設定すると、ZFString の値に基づいて、ZFSize の適切な値が InterSystems IRIS によって自動的に計算されます。式は以下のとおりです。

$$\text{ZFSize} = (1 \text{ 文字あたりのバイト数} * \text{ZFString}) + 2050$$

例

ZFString に既定値の 32,767 を使用する場合、以下ようになります。

- Unicode を使用している場合、1 文字は 2 バイトです。したがって、ZFSize の計算値は 67584 (= 2 * 32767 + 2050) バイトになります。
- UNIX® では、1 文字は 4 バイトです。したがって、ZFSize の計算値は 133118 (= 4 * 32767 + 2050) バイトになります。

このパラメータの変更

管理ポータル の [\[メモリ詳細\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[メモリ詳細\]](#)) で、[\[ZFString\]](#) または [\[ZFSize\]](#) の値を編集します。このパラメータは 0, 0 に設定することをお勧めします。

管理ポータルを使用する代わりに、Config.config クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)”を参照)、zfheap を変更できます。

この設定を編集した場合、変更内容を適用するには、InterSystems IRIS を再起動する必要があります。

[ConfigFile]

このトピックでは、CPF の [ConfigFile] セクションのパラメータについて説明します。

Version

構成パラメータ・ファイルのバージョンを確認します。

構文

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

n は、最大 7 文字の数値の文字列で、通常はドット形式です。

説明

Version は、構成パラメータ・ファイル (CPF) のバージョン番号です。この値は、製品のリリース番号とは無関係です。製品のアップグレードまたはインストールのときに自動的に更新されます。この番号は、パラメータ・ファイルの構文と構造を表すもので、ファイル内でのパラメータ値の変更を表すものではありません。

重要 この値を既定値から変更すると、InterSystems IRIS® Data Platform が起動に失敗するおそれがあります。ユーザが編集した複数の CPF を区別するには、“構成パラメータ・ファイルの概要” に説明されている “[コメント](#)” を使用してください。

[Databases]

このトピックでは、CPF の [Databases] セクションのパラメータについて説明します。

Database

InterSystems IRIS® Data Platform データベースを定義します。

構文

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

Name は文字列です。a、b、c、d、e、および f の説明は後述します。

説明

構成パラメータ・ファイル (CPF) の [Databases] セクションには、インスタンスに接続されているすべてのデータベースのエントリが含まれます。各エントリには、データベースを定義する、最大 6 つのコンマ区切りの引数を指定できます。最初の引数であるディレクトリ指定のみが必須です。その他の引数は、指定しない場合、既定値が使用されます。各値は以下のとおりです。

- ・ a – データベース・ファイルの完全ディレクトリ指定。必須項目。
- ・ b – リモート・インスタンス名 (ローカル・インスタンスの場合は空白)。インスタンスがリモートの場合は (名前が空白以外)、以降のフィールドは無視されます。既定はローカル・インスタンスです。
- ・ c – 起動時にデータベースのマウントが必要かどうかを指定します (1 または 0)。有効な場合は、起動時にデータベースをマウントする必要があります。マウントしないと起動に失敗します。既定は 0 です (起動時にデータベースをマウントしません)。
- ・ d – この値は無視されるので、空白のままにしてください。
- ・ e – この値は無視されるので、空白のままにしてください。
- ・ f – ストリーム位置。このデータベースと関連付けられたストリームの配置先のディレクトリです。既定は空白 (") です (推奨)。この既定位置は、データベース・ディレクトリの下の子ディレクトリ **stream** です。例えば、**c:¥abc** にあるデータベースの場合、既定のストリーム位置は **c:¥abc¥stream** です。

既定では、上記の各設定を指定していなくても、IRISSYS、IRISLIB、IRISTEMP、および IRISAUDIT の各データベースは、インスタンスに対してローカルであること、および起動時にマウントされることがそれぞれ必要です。

例

[Databases] セクションでは、各行が引数 *Name=a,b,c,d,e,f* を持つデータベースです。以下は Windows の場合の例です。

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

ミラーリングなしのリモート・データベースの例：

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


ミラーリングありのリモート・データベースの例 : このミラーリングありの例では、2 つのリモート・データベース (ミラーリングされるデータベースとミラーリングされないデータベースの両方) は、ミラーリングなしの環境の場合とは異なる方法でフォーマットされています。

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

このパラメータの変更

管理ポータルの **[ローカルデータベース]** ページ ([システム管理] > [構成] > [システム構成] > [ローカルデータベース]) に移動します。新しいエントリを追加するには、**[新規データベース作成]** を選択します。既存のエントリを編集するには、そのエントリの行の **[編集]** を選択します。

[Debug]

構成パラメータ・ファイルには、[Debug] セクションが含まれることがあります。[Debug] の設定は、各種診断に使用されます。このトピックでは、構成パラメータ・ファイルの [Debug] セクションに含めることができる 2 つのパラメータについて説明します。その他すべての [Debug] 設定はインターシステムズの占有情報であるため、このドキュメントでは説明しません。

Dumpstyle

コア・ダンプのスタイルを指定します。

構文

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

n は、0 から 4 の範囲の整数です。既定値は 3 です。

説明

InterSystems IRIS® Data Platform によりコア・ダンプが実行される際、このオプションを使用してダンプのスタイルを設定できます。以下は、値とその意味です。

注釈 UNIX® では、すべてのダンプ・スタイルでコア・ファイルが生成されます。終了前にプロセス自体により、クリーンアップが可能な限り実行されます。

- ・ 0
 - Windows では、これは pid.dmp ファイルです。
- ・ 1
 - Windows では、これは irisfpid.dmp という名前の Windows minidump ファイル (type = MiniDumpWithFullMemory) であり、このファイルは WinDbg (Microsoft が提供するデバグガ) で読み取り可能です。これは最も包括的なダンプ・オプションですが、このオプションでは、非常に大きなダンプ・ファイルが作成されることがあります。
- ・ 2
 - Windows では、これはプロセスが最小限のクリーンアップ (deqallresources と GRETRELEASE) を実行した後、例外を再送信する、古いスタイルの例外処理です。この目的は、デバグガで例外を検出し、分析のためにできる限りの情報を残すことです。
 - UNIX® では、これは、コア・ファイルに共有メモリ領域が含まれないように、中止の前に共有メモリのアタッチを解除します。
- ・ 3
 - Windows では、これは、irisipid.dmp という名前の Windows minidump ファイル (type = MiniDumpWithDataSegs | MiniDumpWithPrivateReadWriteMemory | MiniDumpWithIndirectlyReferencedMemory) であり、このファイルは WinDbg で読み取り可能です。このオプションでは、かなり大きいながらも有用なダンプ・ファイルが作成されます。dumpstyle が指定されていない場合は、これが Windows 上の新しい既定値です。
- ・ 4
 - Windows では、これは irismpid.dmp という名前の Windows minidump ファイル (type = MiniDumpNormal) であり、このファイルは WinDbg で読み取り可能です。このオプションでは、最低限の情報が含まれた小さいダンプ・ファイルが作成されます。

アクティブな値は、`$system.Config.ModifyDumpStyle(NewValue)` を使用して変更できます。これにより、すべての新規 InterSystems IRIS プロセスの値が変更されます。iris.cpf 内の値は変更されません。

このパラメータの変更

テキスト・エディタで CPF を編集して Dumpstyle を変更できます (“[アクティブな CPF の編集](#)” を参照)。CPF にまだ [Debug] セクションが含まれていない場合は、自分で追加する必要があります。以下に例を示します。

```
[ConfigFile]
...

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

[Databases]
...
```

Semsperset

各セットに割り当てるセマフォの数を設定します。

構文

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

n は整数です。最大値はありません。既定値は 0 です。

説明

Semsperset は、InterSystems IRIS® データ・プラットフォームが各セマフォ・セットに割り当てるセマフォの数です。このパラメータが 0 (既定値) に設定されている場合、InterSystems IRIS は各セットに最大数のセマフォを割り当てます。その結果、セマフォ・セットの数は最小になります。

semsperset の値を大きくすると、セマフォ・セットのサイズが大きくなり、数は少なくなります。各セットのセマフォ数を少なくすると、特に Linux システムでは (例えば、semsperset を 250 に設定)、パフォーマンスが向上する可能性があります。ただし、これは十分にテストされていません。

詳細は、“[インターシステムズ製品のセマフォ](#)”を参照してください。

このパラメータの変更

テキスト・エディタで CPF を編集して semsperset を変更できます (“[アクティブな CPF の編集](#)”を参照)。CPF にまだ [Debug] セクションが含まれていない場合は、自分で追加する必要があります。以下に例を示します。

```
[ConfigFile]
...

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

[Databases]
...
```

[Devices]

このトピックでは、CPF の [Devices] セクションのパラメータについて説明します。

Devices

デバイス・タイプを定義します。

構文

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

説明

構成パラメータ・ファイル (CPF) の [Devices] セクションには、InterSystems IRIS® Data Platform によって検出されたすべてのデバイスのエントリが含まれます。各エントリには、Name (定義されたデバイスのタイトルまたは番号) と、デバイスを定義する、キャレット文字 (^) で区切られた 8 個の文字列があります。すべての文字列の最大長は 128 文字です。ただし、Description (g) の文字列のみ最大長は 256 文字です。各エントリは以下のとおりです。

- ・ (a) PhysicalDevice — このデバイスを参照するために使用する物理名。PhysicalDevice には、このデバイスの [OPEN](#) コマンドで使用する device 引数を指定します。この名前は最大 128 文字の英数字で構成することができます。空白文字も使用できます。例えば、プリンタの場合、以下のように入力します。MYNAME はコンピュータ名です。

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

または、以下ようになります。

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

- ・ (b) Type — デバイスのタイプ。次のオプションがあります。TRM= ターミナル。SPL= スプール・デバイス。MT= 磁気テープ・ドライブ。BT= カートリッジ・テープ・ドライブ。OTH= プリンタやシーケンシャル・ファイルなどのその他のデバイス。既定値はデバイスのタイプによって異なります。
- ・ (c) SubType — デバイス・サブタイプの定義を調整するための設定。サブタイプによって、ターミナルの特性を指定します。サブタイプは、デバイスに適切な OPEN コマンドを作成するために使用します。サブタイプ情報はターミナル・タイプごとになります。
- ・ (d) Prompt — プロンプトを選択します。有効な入力、1、2、または NULL (空白または空のフィールド) です。1 は、**[現デバイスの場合はこのデバイスを自動使用]** に対応します。2 は、**[定義済み設定でこのデバイスを自動使用]** に対応します (事前定義された [右マージン] 設定と [パラメータ] 設定)。NULL は、**[デバイスプロンプト表示]** に対応します (既定デバイスが定義されたデバイス選択プロンプトが表示されます)。
- ・ (e) OpenParameter — このデバイスの [OPEN](#) コマンドの引数である parameters、timeout、および mnespace を指定する、コロン区切りの文字列。OpenParameter の文字列の構文は以下のとおりです。

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

parameters を指定する括弧内には、個々の項目をコロンで区切って次のように入力します。

```
param1:param2:param3
```

その結果、次のようになります。

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

timeout と mnespace はオプションですが、指定する場合は、適切な数のコロンを使用して、OpenParameter の文字列内の先行する入力内容と区切る必要があります。

parameters は、指定するパラメータが複数ある場合にのみ、括弧で囲む必要があります。指定するパラメータが何もない場合や 1 つしかない場合は、括弧を文字列から削除してかまいません。したがって、次の文字列は、OpenParameter の正しく完全な文字列です。

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


mnespace を指定する場合は、上記のように二重引用符で囲む必要があります。

さまざまな構文の例など、[OPEN](#) コマンドとその引数の詳細は、“[ObjectScript リファレンス](#)”を参照してください。

- (f) AlternateDevice – 別のデバイスのデバイス ID。AlternateDevice に入力する値は、別のデバイスに設定された Name などの定義されたニーモニックでなければなりません。

 デバイスに対して AlternateDevice 値を指定すると、%IS ユーティリティのユーザが、“A”と指定することによって、InterSystems IRIS での代替デバイスの使用を指定できるようになります。%IS は、文字ベース・アプリケーションの汎用的なデバイス選択ユーティリティです。%IS の詳細は、“[%IS の使用](#)”を参照してください。特に、“[%IS Mnemonics](#)”のセクションでは、%IS の“A”コードの入力に関する規約について説明しています。
- (g) Description – デバイスの場所を示すテキストの説明。このフィールドは、構成するマシンをユーザが把握するためにあります。
- (h) Alias – このデバイスの代替デバイス ID (番号)。エイリアスはすべて一意である必要があります。この値は、[OPEN](#) コマンドの device 引数として使用できます。

例

[Devices] セクションの中で、各エントリ Name=a^b^c^d^e^f^g^h は全体が 1 行で記述されます。

```
[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^
```

このパラメータの変更

管理ポータルの [\[デバイス\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[デバイス設定\]](#) > [\[デバイス\]](#)) は、既存のデバイスのリストです。[\[新規デバイス作成\]](#)、[\[編集\]](#)、または [\[削除\]](#) を選択して、リストを変更します。

[DeviceSubTypes]

このトピックでは、構成パラメータ・ファイルの [DeviceSubTypes] セクションのパラメータについて説明します。

DeviceSubTypes

デバイス・サブタイプを定義します。

構文

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

説明

構成パラメータ・ファイル (CPF) の [DeviceSubTypes] セクションには、このインストールで構成されている各サブタイプのエントリが含まれます。各エントリには、Name と、デバイス・サブタイプを定義する、キャレット文字 (^) で区切られた 9 個の値があります。各エントリは以下のとおりです。

- ・ (a) RightMargin – 右マージンの位置を示す数値。デバイス出力は、この文字数で折り返されます。
- ・ (b) FormFeed – #,\$C(code1,code2...) の形式で選択デバイスの改ページを示す ASCII コード。この設定は InterSystems IRIS® Data Platform CHUI ユーティリティで使用されます。
- ・ (c) ScreenLength – デバイスの 1 画面または 1 ページを構成する行数。
- ・ (d) Backspace – \$C(code1) の形式で選択デバイスのバックスペース文字を示す ASCII コード。この設定は InterSystems IRIS CHUI ユーティリティで使用されます。
- ・ (e) CursorControl – \$C(code1) の形式で選択デバイスのカーソルを示す ASCII コード。
- ・ (f) EraseEOL – \$C(code1,code2) の形式でこのデバイスの EOL (End of Line) 文字の消去を示す ASCII コード。
- ・ (g) EraseEOF – \$C(code1,code2...) の形式で選択デバイスの EOF (End of File) 文字の消去を示す ASCII コード。
- ・ (h) ZU22FormFeed – \$C(code1,code2) の形式で選択デバイスの改ページを示す ASCII コード。この設定は InterSystems ターミナルの出力で使用されます。
- ・ (i) ZU22Backspace – \$C(code1) の形式で選択デバイスのバックスペースを示す ASCII コード。この設定は InterSystems ターミナルの出力で使用されます。

既定値はデバイス・タイプによって異なります。

例

[DeviceSubTypes] セクションの例を以下に示します。この例では、長い行がページにちょうど収まるように折り返されています。 .cpf ファイル自体では、各エントリは全体が 1 行で記述されます。

```
[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)^$C(27,91,72,27,91,74)^$C(8,32,8)
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)^$C(27,91,72,27,91,74)^$C(8,32,8)
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)^$C(27,91,72,27,91,74)^$C(8,32,8)
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)^^^^^
```

このパラメータの変更

管理ポータルの **[デバイスサブタイプ]** ページ ([システム管理] > [構成] > [デバイス設定] > [デバイスサブタイプ]) は、既存のサブタイプのリストです。**[新規サブタイプ作成]**、**[編集]**、または **[削除]** を選択して、リストを変更します。

[ECP]

このトピックでは、CPF の [ECP] セクションのパラメータについて説明します。

ClientReconnectDuration

ECP 再接続を試行する期間を設定します。

構文

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

n は、10 から 65,636 (秒) の範囲の整数です。既定値は 1,200 です。

説明

ClientReconnectDuration は、アプリケーション・サーバ (ECP クライアント) が、接続の再確立を試行し続ける期間 (秒数) です。この期間を経過すると接続不能と判断されるか、接続の失敗が宣言されます。アプリケーション・サーバ (ECP クライアント) は、ClientReconnectDuration がタイムアウトになるまで、[ClientReconnectInterval](#) で設定された間隔で接続を再試行します。既定値の 1200 は 20 分に相当します。

このパラメータの変更

管理ポータルの [ECP 設定] ページ ([システム管理] > [構成] > [接続性] > [ECP 設定]) で、[このシステムを ECP アプリケーションサーバとする] 列の [リカバリまでの待機時間] を編集します。

管理ポータルを使用する代わりに、Config.ECP クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、ClientReconnectDuration を変更できます。

ClientReconnectInterval

ECP 再接続を試行する間隔を設定します。

構文

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

n は、1 から 60 (秒) の範囲の整数です。既定値は 5 です。

説明

ClientReconnectInterval は、データ・サーバ (ECP サーバ) が使用不可能になったときに、再接続を試行するまで待機する時間 (秒数) です。アプリケーション・サーバ (ECP クライアント) は、[ClientReconnectDuration](#) がタイムアウトになるまで、ClientReconnectInterval で設定された間隔で接続を再試行します。

このパラメータの変更

管理ポータル の [ECP 設定] ページ ([システム管理] > [構成] > [接続性] > [ECP 設定]) で、[このシステムを ECP アプリケーションサーバとする] 列の [再接続までの時間] を編集します。

管理ポータルを使用する代わりに、Config.ECP クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、ClientReconnectInterval を変更できます。

ServerTroubleDuration

ECP 接続が障害状態を続ける時間 (秒数) を設定します。

構文

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

n は、20 から 65,636 (秒) の範囲の整数です。既定値は 60 です。

説明

ServerTroubleDuration は、ECP 接続の障害状態が続く時間 (秒数) です。この時間が経過すると、データ・サーバ (ECP サーバ) は、接続が回復不可能でリカバリできないと判断します。

このパラメータの変更

管理ポータル の [ECP 設定] ページ ([システム管理] > [構成] > [接続性] > [ECP 設定]) で、[このシステムを ECP データサーバとする] 列の [トラブル状態の時間間隔] を編集します。

管理ポータルを使用する代わりに、Config.ECP クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、ServerTroubleDuration を変更できます。

[ECPServers]

このトピックでは、CPF の [ECPServers] セクションのサーバ・エントリについて説明します。

ECPServers

ECP サーバを定義します。

構文

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

説明

[ECPServers] のエントリは、接続先とする ECP サーバのアドレスとポート、および ECP サーバがミラー・プライマリである場合に接続をリダイレクトする方法を定義します(データ・サーバとしてのミラー・プライマリの構成に関する重要な情報は、“[ミラーへのアプリケーション・サーバ接続の構成](#)”を参照してください)。

- ・ Address – 接続先とする ECP サーバのアドレス。
- ・ Port – 接続先とする ECP サーバのポート番号。
- ・ MirrorConnection – ミラー・プライマリに接続する際の動作。既定値は、データ・サーバがミラー・メンバでないことを示す 0 (または空白) です。値 1 は、ECP サーバがミラー・フェイルオーバー・メンバであることを示し、フェイルオーバーの場合にはプライマリであるいずれかのメンバにミラー接続がリダイレクトされます。値 -1 は、ECP サーバがフェイルオーバー・メンバまたは DR 非同期のいずれかであることを示し、ミラー接続はその特定の ECP サーバに制限されます。この ECP サーバがバックアップ・メンバになった場合、このサーバはプライマリになるまで接続を受け入れません。
- ・ SSLConfig – 値が 1 の場合、サーバの接続に TLS/SSL が使用されます。
- ・ BatchMode – 値が 1 の場合、このデータ・サーバのサーバ・プロセスはバッチ・モードで実行されます。バッチ・モードでは、データ・サーバは常にブロックをロードして、それらをバッチ・レベルでキャッシュします。

このパラメータの変更

管理ポータル の [ECP 設定] ページ ([システム管理] > [構成] > [接続性] > [ECP 設定]) は、ECP データ・サーバのリストです。[新規リモートデータサーバの追加] を選択して、新しい ECP データ・サーバを追加します。

注釈 管理ポータルから MirrorConnection プロパティを -1 に設定することはできません。

管理ポータルを使用する代わりに、Config.ECPServers クラス (クラス・リファレンスを参照) を使用して、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)”を参照)、ECPServers を変更できます。

[Gateways]

ここでは、CPF の [Gateways] セクションについて説明します。このセクションには、InterSystems IRIS® データ・プラットフォームのインスタンスとそのホスト・システム上で構成された各ゲートウェイに関するエントリがあります。ゲートウェイは外部言語サーバとも呼ばれ、InterSystems IRIS と外部言語プラットフォームとの間に全面的に統合した双方向接続を提供します。

各定義は、その名前の後に = 文字と、ゲートウェイを定義するコンマ区切りリストの値を記述して構成されています。必須のフィールドもあれば、空白のままとすることができるフィールドもあり、その場合は既定値が使用されます。以下に示すように、新たにインストールしたインスタンスの既定の CPF には、7 種類のゲートウェイとして、.NET、IntegratedML、JDBC、Java、Python、R、XSLT が定義されています。このような種類のローカル・ゲートウェイのほか、リモート・システム上のリモート・ゲートウェイを定義できます。

```
[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
```

注釈 JDBC、Java、IntegratedML、XSLT、R の各ゲートウェイはすべて Java ベースであり、その定義には同じフィールドを使用します。

Type フィールドと Port フィールドはすべての種類のゲートウェイで必須です。3 番目のフィールドである Resource では、ゲートウェイを使用するユーザが USE 権限を必須とする [ゲートウェイ・リソース](#) を指定します。例えば、上記の既定の %JDBC Server 定義では、JDBC ゲートウェイを使用するユーザが **%Gateway_SQL:USE** 特権を必要とすることを指定しています。適切なリソースの指定によってすべてのゲートウェイを保護することを強くお勧めします。

%Admin_ExternalLanguageServerEdit:USE 特権があれば、以下の方法で任意の種類の新しいゲートウェイを作成でき、既存のゲートウェイを編集または削除できます。この特権は、既定で **%Manager** ロールが保持しています。

- ・ 管理ポータル の [外部言語サーバ] ページ ([システム管理]→[構成]→[接続性]→[外部言語サーバ])
- ・ Config.Gateway API
- ・ インスタンスの CPF を、[iris merge](#) で変更するか、[直接編集](#) する

[外部言語サーバ] ページまたは \$system.external インタフェースを使用すると、ゲートウェイを起動して、そのアクティビティ・ログを表示できます。

既定の CPF またはユーザによる操作で定義されたゲートウェイの名前は変更できません。

%DotNet Server

.NET ゲートウェイを定義します。

構文

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

説明

%DotNet Server は既定の CPF に記述された .NET ゲートウェイ定義の名前です。これとは別の .NET ゲートウェイを別の名前で作成できるほか、“[Gateways]”の説明にあるように既存のゲートウェイの編集または削除もできます。各定義は、その名前の後に = 文字と、ゲートウェイを定義するコンマ区切りリストの値を記述して構成されています。必須のフィールドもあれば、空白のままとすることができるフィールドもあり、その場合は既定値が使用されます。以降の 2 つのセクションでは、.NET ゲートウェイに固有のフィールドと、すべてのゲートウェイで利用できるフィールドについて説明しています。詳細は、“.NET 外部サーバのセットアップ” および “.NET 用の外部サーバ構成の定義” を参照してください。

.NET ゲートウェイの各フィールド

.NET ゲートウェイ定義にある最初の 3 つのフィールドはすべてのゲートウェイに共通であり、続く 3 つのフィールドは .NET ゲートウェイに固有です。これらは以下の順序で記述されています。

- Type (必須) – ゲートウェイのタイプ。.NET ゲートウェイを定義すると .NET に設定されます。
- Port (必須) – InterSystems IRIS でゲートウェイとプロキシ・クラスとの通信に使用する TCP ポート番号。このポート番号は、サーバ上のその他のローカル TCP ポートと競合していないことが必要です。
- Resource – このゲートウェイへのアクセスを制御する[ゲートウェイ・リソース](#)。既定の CPF では %Gateway_Object です。

重要 リソースを指定していない場合、ゲートウェイはパブリックですが、この機能を使用してすべてのゲートウェイを保護することを強くお勧めします。

- DotNetVersion – 使用する .NET バージョンを指定します。指定できる値は N6.0、N5.0、F4.6.2、F4.5、F3.5、F2.0、C2.1 です (既定値は N6.0)。

注釈 既定値が F4.5 である前のバージョンからインスタンスをアップグレードしたときに N6.0 が設定されるように、既定の CPF では既定値として N6.0 が指定されています。

- FilePath – ゲートウェイの実行可能ファイルの場所。ローカル・サーバ上でゲートウェイを開始するコマンドをアセンブルするときに使用します。この設定を指定していない場合、既定のディレクトリである install-dir¥dev¥dotnet¥bin¥が使用され、DotNetVersion の設定に従って適切なサブディレクトリが選択されます。例えば、DotNetVersion が N6.0 であれば、FilePath は install-dir¥dev¥dotnet¥bin¥net6.0 になります。

注釈 InterSystems IRIS の現在のバージョンでは、DotNetVersion の値 F4.5、F2.0、C2.1 に対応する実行可能ファイルはインストールされません。以前のバージョンの実行可能ファイルを使用できるように、これらの値が用意されています。

- Exec32 – 64 ビット・プラットフォームでこの値を 1 に設定すると、ゲートウェイが 32 ビットとして実行されます。既定値は 0 で、実行は 64 ビットになります。

一般ゲートウェイの各フィールド

残りの 8 つのフィールドはどのゲートウェイ定義でも使用できます。これらは以下の順序で記述されています。

- ・ `SSLConfigurationServer` – サーバの TLS/SSL に使用する SSL/TLS 構成の名前。
- ・ `SSLConfigurationClient` – クライアントの TLS/SSL に使用する SSL/TLS 構成の名前。
- ・ `VerifySSLHostName` – TLS/SSL クライアントでホスト名を検証するかどうかの指定。既定値は 0 で、ホスト名は検証されません。
- ・ `UseSharedMemory` – 共有メモリが使用できる場合に、接続に共有メモリを使用するかどうかの指定。既定値は 1 (共有メモリを使用) です。
- ・ `LogFile` – InterSystems IRIS とゲートウェイ間のすべての通信をログ記録するファイルの完全パス名。このオプションのプロパティは、デバッグ時にのみ使用してください。最大長は 1023 文字です。
- ・ `AllowedIpAddresses` – 受信接続を許可する IP アドレス。すべてのローカル IP アドレス (127.0.0.1、VPN アドレスなど) での接続をマシンに許可する場合は 0.0.0.0 を指定します。既存の単一ローカル IP アドレスを指定して、接続をその IP アドレスのみに制限することもできます。既定値は、127.0.0.1 です。
- ・ `ConnectionTimeout` – ゲートウェイとの接続が確立するまでに待機する秒数。範囲は 2 ~ 300 です。既定値は 5 です。
- ・ `InitializationTimeout` – ゲートウェイの初期化中に応答を待機する秒数。範囲は 2 ~ 300 です。既定値は 5 です。

このパラメータの変更

ゲートウェイ定義の変更と追加の詳細は、“[\[Gateways\]](#)” を参照してください。

%IntegratedML Server

IntegratedML ゲートウェイを定義します。

構文

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

説明

%IntegratedML Server は既定の CPF に記述された IntegratedML ゲートウェイ定義の名前です。これとは別の IntegratedML ゲートウェイを別の名前で作成できるほか、“[\[Gateways\]](#)”の説明にあるように既存のゲートウェイを編集または削除できます。各定義は、その名前の後に = 文字と、ゲートウェイを定義するコンマ区切りリストの値を記述して構成されています。必須のフィールドもあれば、空白のままとすることができるフィールドもあり、その場合は既定値が使用されます。以降の 2 つのセクションでは、IntegratedML ゲートウェイとその他の Java ベース ゲートウェイに固有のフィールドと、すべてのゲートウェイで利用できるフィールドについて説明しています。詳細は、“Java 外部サーバのセットアップ” および “Java 用の外部サーバ構成の定義” を参照してください。

IntegratedML ゲートウェイの各フィールド

IntegratedML ゲートウェイ定義にある最初の 3 つのフィールドはすべてのゲートウェイに共通であり、続く 3 つのフィールドは Java ベース ゲートウェイに固有です。これらは以下の順序で記述されています。

- ・ Type (必須) – ゲートウェイのタイプ。IntegratedML ゲートウェイを定義すると ML に設定されます。
- ・ Port (必須) – InterSystems IRIS でゲートウェイとプロキシ・クラスとの通信に使用する TCP ポート番号。このポート番号は、サーバ上のその他のローカル TCP ポートと競合していない必要があります。
- ・ Resource – このゲートウェイへのアクセスを制御する[ゲートウェイ・リソース](#)。既定の CPF では %Integrated_ML Server の場合に **%Gateway_ML** です。

重要 リソースを指定していない場合、ゲートウェイはパブリックですが、この機能を使用してすべてのゲートウェイを保護することを強くお勧めします。

- ・ ClassPath – Java 仮想マシン (JVM) の起動時に引数として渡す必要があるファイル名です。一般的に、各ファイルにはゲートウェイを介して使用するクラスを記述します。このクラスパスでは正しく引用符を使用し、プラットフォーム固有の正確な区切り文字で各ファイル名を区切る必要があります。
- ・ JVMArgs – JVM に渡すオプションの引数。ゲートウェイを起動するコマンドに使用されます。
- ・ JavaHome – JVM の場所。ゲートウェイを起動するコマンドに使用されます。

重要 ホスト・システムに既定の JVM を設定していない限り、このフィールドは必須です。

一般ゲートウェイの各フィールド

残りの 8 つのフィールドはどのゲートウェイ定義でも使用できます。“[%DotNet Server](#)”の“[一般ゲートウェイの各フィールド](#)”セクションを参照してください。

このパラメータの変更

ゲートウェイ定義の変更と追加の詳細は、“[\[Gateways\]](#)”を参照してください。

%JDBC Server

JDBC ゲートウェイを定義します。

構文

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

説明

%JDBC Server は既定の CPF に記述された JDBC ゲートウェイ定義の名前です。これとは別の JDBC ゲートウェイを別の名前で作成できるほか、“[\[Gateways\]](#)”の説明にあるように既存のゲートウェイの編集または削除もできます。各定義は、その名前の後に = 文字と、ゲートウェイを定義するコンマ区切りリストの値を記述して構成されています。必須のフィールドもあれば、空白のままとすることができるフィールドもあり、その場合は既定値が使用されます。以降の 2 つのセクションでは、JDBC ゲートウェイとその他の Java ベース ゲートウェイに固有のフィールドと、すべてのゲートウェイで利用できるフィールドについて説明しています。詳細は、“Java 外部サーバのセットアップ” および “Java 用の外部サーバ構成の定義” を参照してください。

JDBC ゲートウェイの各フィールド

JDBC ゲートウェイ定義にある最初の 3 つのフィールドはすべてのゲートウェイに共通であり、続く 3 つのフィールドは Java ベース ゲートウェイに固有です。これらは以下の順序で記述されています。

- ・ Type (必須) – ゲートウェイのタイプ。JDBC ゲートウェイを定義すると JDBC に設定されます。
- ・ Port (必須) – InterSystems IRIS でゲートウェイとプロキシ・クラスとの通信に使用する TCP ポート番号。このポート番号は、サーバ上のその他のローカル TCP ポートと競合していない必要があります。
- ・ Resource – このゲートウェイへのアクセスを制御する[ゲートウェイ・リソース](#)。既定の CPF では %JDBC Server の場合に **%Gateway_SQL** です。

重要 リソースを指定していない場合、ゲートウェイはパブリックですが、この機能を使用してすべてのゲートウェイを保護することを強くお勧めします。

- ・ ClassPath – Java 仮想マシン (JVM) の起動時に引数として渡す必要があるファイル名です。一般的に、各ファイルにはゲートウェイを介して使用するクラスを記述します。このクラスパスでは正しく引用符を使用し、プラットフォーム固有の正確な区切り文字で各ファイル名を区切る必要があります。
- ・ JVMArgs – JVM に渡すオプションの引数。ゲートウェイを起動するコマンドに使用されます。
- ・ JavaHome – JVM の場所。ゲートウェイを起動するコマンドに使用されます。

重要 ホスト・システムに既定の JVM を設定していない限り、このフィールドは必須です。

一般ゲートウェイの各フィールド

残りの 8 つのフィールドはどのゲートウェイ定義でも使用できます。“[%DotNet Server](#)”の“[一般ゲートウェイの各フィールド](#)”セクションを参照してください。

このパラメータの変更

ゲートウェイ定義の変更と追加の詳細は、“[\[Gateways\]](#)”を参照してください。

%Java Server

Java ゲートウェイを定義します。

構文

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

説明

%Java Server は既定の CPF に記述された Java ゲートウェイ定義の名前です。これとは別の Java ゲートウェイを別の名前で作成できるほか、“[\[Gateways\]](#)”の説明にあるように既存のゲートウェイの編集または削除もできます。各定義は、その名前の後に = 文字と、ゲートウェイを定義するコンマ区切りリストの値を記述して構成されています。必須のフィールドもあれば、空白のままとすることができるフィールドもあり、その場合は既定値が使用されます。以降の 2 つのセクションでは、Java ゲートウェイとその他の Java ベース ゲートウェイに固有のフィールドと、すべてのゲートウェイで利用できるフィールドについて説明しています。詳細は、“[Java 外部サーバのセットアップ](#)”および“[Java 用の外部サーバ構成の定義](#)”を参照してください。

Java ゲートウェイの各フィールド

Java ゲートウェイ定義にある最初の 3 つのフィールドはすべてのゲートウェイに共通であり、続く 3 つのフィールドは Java ベース ゲートウェイに固有です。これらは以下の順序で記述されています。

- ・ Type (必須) – ゲートウェイのタイプ。Java ゲートウェイを定義すると Java に設定されます。
- ・ Port (必須) – InterSystems IRIS でゲートウェイとプロキシ・クラスとの通信に使用する TCP ポート番号。このポート番号は、サーバ上のその他のローカル TCP ポートと競合していないことが必要です。
- ・ Resource – このゲートウェイへのアクセスを制御する[ゲートウェイ・リソース](#)。既定の CPF では %Java Server の場合に **%Gateway_Object** です。

重要 リソースを指定していない場合、ゲートウェイはパブリックですが、この機能を使用してすべてのゲートウェイを保護することを強くお勧めします。

- ・ ClassPath – Java 仮想マシン (JVM) の起動時に引数として渡す必要があるファイル名です。一般的に、各ファイルにはゲートウェイを介して使用するクラスを記述します。このクラスパスでは正しく引用符を使用し、プラットフォーム固有の正確な区切り文字で各ファイル名を区切る必要があります。
- ・ JVMArgs – JVM に渡すオプションの引数。ゲートウェイを起動するコマンドに使用されます。
- ・ JavaHome – JVM の場所。ゲートウェイを起動するコマンドに使用されます。

重要 ホスト・システムに既定の JVM を設定していない限り、このフィールドは必須です。

一般ゲートウェイの各フィールド

残りの 8 つのフィールドはどのゲートウェイ定義でも使用できます。“[%DotNet Server](#)”の“[一般ゲートウェイの各フィールド](#)”セクションを参照してください。

このパラメータの変更

ゲートウェイ定義の変更と追加の詳細は、“[\[Gateways\]](#)”を参照してください。

%Python Server

Python ゲートウェイを定義します。

構文

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

説明

%Python Server は既定の CPF に記述された Python ゲートウェイ定義の名前です。これとは別の Python ゲートウェイを別の名前で作成できるほか、“[\[Gateways\]](#)” の説明にあるように既存のゲートウェイの編集または削除もできます。各定義は、その名前の後に = 文字と、ゲートウェイを定義するコンマ区切りリストの値を記述して構成されています。必須のフィールドもあれば、空白のままとすることができるフィールドもあり、その場合は既定値が使用されます。以降の 2 つのセクションでは、Python ゲートウェイに固有のフィールドと、すべてのゲートウェイで利用できるフィールドについて説明しています。詳細は、“Python 外部サーバのセットアップ” および “Python 用の外部サーバ構成の定義” を参照してください。

Python ゲートウェイの各フィールド

Python ゲートウェイ定義にある最初の 3 つのフィールドはすべてのゲートウェイに共通であり、続く 3 つのフィールドは Python ゲートウェイに固有です。これらは以下の順序で記述されています。

- Type (必須) – ゲートウェイのタイプ。Python ゲートウェイを定義すると Python に設定されます。
- Port (必須) – InterSystems IRIS でゲートウェイとプロキシ・クラスとの通信に使用する TCP ポート番号。このポート番号は、サーバ上のその他のローカル TCP ポートと競合していないことが必要です。
- Resource – このゲートウェイへのアクセスを制御する[ゲートウェイ・リソース](#)。既定の CPF では %Python Server の場合に **%Gateway_SQL** です。

重要 リソースを指定していない場合、ゲートウェイはパブリックですが、この機能を使用してすべてのゲートウェイを保護することを強くお勧めします。

- PythonPath – Python インタプリタの場所。ローカル・サーバ上でゲートウェイを開始するコマンドをアセンブルするときに使用します。

重要 ホスト・システムに既定の Python インタプリタを設定していない限り、このフィールドは必須です。

- PythonOptions – ローカル・サーバ上でゲートウェイを開始するコマンドで指定する Python のオプションです。
- (未使用) – このフィールドは空のままにします (コンマの連続になります)。

一般ゲートウェイの各フィールド

残りの 8 つのフィールドはどのゲートウェイ定義でも使用できます。“[%DotNet Server](#)” の “[一般ゲートウェイの各フィールド](#)” セクションを参照してください。

このパラメータの変更

ゲートウェイ定義の変更と追加の詳細は、“[\[Gateways\]](#)” を参照してください。

%R Server

R ゲートウェイを定義します。

構文

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

説明

%R Server は既定の CPF に記述された R ゲートウェイ定義の名前です。これとは別の R ゲートウェイを別の名前で作成できるほか、“[\[Gateways\]](#)”の説明にあるように既存のゲートウェイの編集または削除もできます。各定義は、その名前の後に = 文字と、ゲートウェイを定義するコンマ区切りリストの値を記述して構成されています。必須のフィールドもあれば、空白のままとすることができるフィールドもあり、その場合は既定値が使用されます。以降の 2 つのセクションでは、R ゲートウェイとその他の Java ベース ゲートウェイに固有のフィールドと、すべてのゲートウェイで利用できるフィールドについて説明しています。詳細は、“[Java 外部サーバのセットアップ](#)” および “[Java 用の外部サーバ構成の定義](#)”を参照してください。

R ゲートウェイの各フィールド

R ゲートウェイ定義にある最初の 3 つのフィールドはすべてのゲートウェイに共通であり、続く 3 つのフィールドは Java ベース ゲートウェイに固有です。これらは以下の順序で記述されています。

- Type (必須) – ゲートウェイのタイプ。R ゲートウェイを定義すると R に設定されます。
- Port (必須) – InterSystems IRIS でゲートウェイとプロキシ・クラスとの通信に使用する TCP ポート番号。このポート番号は、サーバ上のその他のローカル TCP ポートと競合していないことが必要です。
- Resource – このゲートウェイへのアクセスを制御する[ゲートウェイ・リソース](#)。既定の CPF では %R Server の場合に **%Gateway_Object** です。

重要 リソースを指定していない場合、ゲートウェイはパブリックですが、この機能を使用してすべてのゲートウェイを保護することを強くお勧めします。

- ClassPath – Java 仮想マシン (JVM) の起動時に引数として渡す必要があるファイル名です。一般的に、各ファイルにはゲートウェイを介して使用するクラスを記述します。このクラスパスでは正しく引用符を使用し、プラットフォーム固有の正確な区切り文字で各ファイル名を区切る必要があります。
- JVMArgs – JVM に渡すオプションの引数。ゲートウェイを起動するコマンドに使用されます。
- JavaHome – JVM の場所。ゲートウェイを起動するコマンドに使用されます。

重要 ホスト・システムに既定の JVM を設定していない限り、このフィールドは必須です。

一般ゲートウェイの各フィールド

残りの 8 つのフィールドはどのゲートウェイ定義でも使用できます。“[%DotNet Server](#)”の“[一般ゲートウェイの各フィールド](#)”セクションを参照してください。

このパラメータの変更

ゲートウェイ定義の変更と追加の詳細は、“[\[Gateways\]](#)”を参照してください。

%XSLT Server

XSLT ゲートウェイを定義します。

構文

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

説明

%XSLT Server は既定の CPF に記述された XSLT ゲートウェイ定義の名前です。これとは別の XSLT ゲートウェイを別の名前で作成できるほか、“[\[Gateways\]](#)”の説明にあるように既存のゲートウェイの編集または削除もできます。各定義は、その名前の後に = 文字と、ゲートウェイを定義するコンマ区切りリストの値を記述して構成されています。必須のフィールドもあれば、空白のままとすることができるフィールドもあり、その場合は既定値が使用されます。以降の 2 つのセクションでは、XSLT ゲートウェイとその他の Java ベース ゲートウェイに固有のフィールドと、すべてのゲートウェイで利用できるフィールドについて説明しています。詳細は、“Java 外部サーバのセットアップ” および “Java 用の外部サーバ構成の定義” を参照してください。

XSLT ゲートウェイの各フィールド

XSLT ゲートウェイ定義にある最初の 3 つのフィールドはすべてのゲートウェイに共通であり、続く 3 つのフィールドは Java ベース ゲートウェイに固有です。これらは以下の順序で記述されています。

- ・ Type (必須) – ゲートウェイのタイプ。XSLT ゲートウェイを定義すると XSLT に設定されます。
- ・ Port (必須) – InterSystems IRIS でゲートウェイとプロキシ・クラスとの通信に使用する TCP ポート番号。このポート番号は、サーバ上のその他のローカル TCP ポートと競合していない必要があります。
- ・ Resource – このゲートウェイへのアクセスを制御する[ゲートウェイ・リソース](#)。既定の CPF では %XSLT Server の場合に **%Gateway_Object** です。

重要 リソースを指定していない場合、ゲートウェイはパブリックですが、この機能を使用してすべてのゲートウェイを保護することを強くお勧めします。

- ・ ClassPath – Java 仮想マシン (JVM) の起動時に引数として渡す必要があるファイル名です。一般的に、各ファイルにはゲートウェイを介して使用するクラスを記述します。このクラスパスでは正しく引用符を使用し、プラットフォーム固有の正確な区切り文字で各ファイル名を区切る必要があります。
- ・ JVMArgs – JVM に渡すオプションの引数。ゲートウェイを起動するコマンドに使用されます。
- ・ JavaHome – JVM の場所。ゲートウェイを起動するコマンドに使用されます。

重要 ホスト・システムに既定の JVM を設定していない限り、このフィールドは必須です。

一般ゲートウェイの各フィールド

残りの 8 つのフィールドはどのゲートウェイ定義でも使用できます。“[%DotNet Server](#)”の“[一般ゲートウェイの各フィールド](#)”セクションを参照してください。

このパラメータの変更

ゲートウェイ定義の変更と追加の詳細は、“[\[Gateways\]](#)”を参照してください。

リモート・サーバ

リモート・ゲートウェイを定義します。

構文

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

説明

既定の CPF の [Gateways] セクションにリモート・サーバは定義されていませんが、リモート・ゲートウェイを作成できるほか、“[Gateways]” の説明にあるように既存のゲートウェイの編集または削除もできます。各定義は、その名前の後に = 文字と、ゲートウェイを定義するコンマ区切りリストの値を記述して構成されています。必須のフィールドもあれば、空白のままとすることができるフィールドもあり、その場合は既定値が使用されます。以降の 2 つのセクションでは、リモートゲートウェイに固有のフィールドと、すべてのゲートウェイで使用できるフィールドについて説明しています。

リモート・ゲートウェイの各フィールド

リモートゲートウェイ定義にある最初の 3 つのフィールドはすべてのゲートウェイに共通であり、続く 3 つのフィールドはリモートゲートウェイに固有です。これらは以下の順序で記述されています。

- ・ Type (必須) – ゲートウェイのタイプ。リモート・ゲートウェイを定義すると `remote` に設定されます。
- ・ Port (必須) – InterSystems IRIS でゲートウェイとプロキシ・クラスとの通信に使用する TCP ポート番号。このポート番号は、リモート・サーバ上のその他のローカル TCP ポートと競合していない必要があります。
- ・ Resource – このゲートウェイへのアクセスを制御する [ゲートウェイ・リソース](#)。管理ポータルを使用して作成するリモート・ゲートウェイの場合 (“[Gateways]” を参照)、既定値は `%Gateway_Object` です。

重要 リソースを指定していない場合、ゲートウェイはパブリックですが、この機能を使用してすべてのゲートウェイを保護することを強くお勧めします。

- ・ Address – ゲートウェイの定義先であるリモート・システムの IP アドレスまたはホスト名です。
- ・ (未使用) – このフィールドは空のままにします (コンマの連続になります)。
- ・ (未使用) – このフィールドは空のままにします (コンマの連続になります)。

一般ゲートウェイの各フィールド

残りの 8 つのフィールドはどのゲートウェイ定義でも使用できます。“%DotNet Server” の “[一般ゲートウェイの各フィールド](#)” セクションを参照してください。

このパラメータの変更

ゲートウェイ定義の変更と追加の詳細は、“[Gateways]” を参照してください。

[IO]

CPF の [IO] セクションのパラメータでは、WRITE コマンドの既定のニーモニック (^%X364) を変更できます。これらの設定は、管理ポータル の [IO 設定] ページにもあります ([システム管理] > [構成] > [デバイス設定] > [IO 設定])。

Other

ターミナルまたはシーケンシャル・ファイル以外のデバイス・タイプに対する WRITE コマンドの既定のニーモニックを設定します。

構文

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

n は、InterSystems IRIS® データ・プラットフォーム・ルーチン名です。既定値は ^%X364 です。

説明

Other 設定では、ターミナルまたはシーケンシャル・ファイル以外のデバイス・タイプに対する既定のニーモニックを指定します。[OPEN](#) コマンドまたは [USE](#) コマンドにニーモニック空間の引数が含まれない場合、InterSystems IRIS はこのデバイス・タイプ用の既定のニーモニックを使用します。詳細は、“[ニーモニック空間によるデバイス制御](#)”を参照してください。

このパラメータの変更

管理ポータルでの [\[IO 設定\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[デバイス設定\]](#) > [\[IO 設定\]](#)) で、[\[Other\]](#) の行に InterSystems IRIS ルーチン名を入力します。

管理ポータルを使用する代わりに、Config.IO クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)”を参照)、Other を変更できます。

File

シーケンシャル・ファイルに対する WRITE コマンドの既定のニーモニックを設定します。

構文

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

n は、InterSystems IRIS® データ・プラットフォーム・ルーチン名です。既定値は ^%X364 です。

説明

File 設定では、シーケンシャル・ファイルに対する既定のニーモニックを指定します。[OPEN](#) コマンドまたは [USE](#) コマンドにニーモニック空間の引数が含まれない場合、InterSystems IRIS はこのデバイス・タイプ用の既定のニーモニックを使用します。詳細は、“[ニーモニック空間によるデバイス制御](#)” を参照してください。

このパラメータの変更

管理ポータル の [\[IO 設定\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[デバイス設定\]](#) > [\[IO 設定\]](#)) で、[\[File\]](#) の行に InterSystems IRIS ルーチン名を入力します。

管理ポータルを使用する代わりに、Config.IO クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、File を変更できます。

Terminal

ターミナル・デバイスに対する WRITE コマンドの既定のニーモニックを設定します。

構文

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

n は、InterSystems IRIS® データ・プラットフォーム・ルーチン名です。既定値は ^%X364 です。

説明

Terminal 設定では、ターミナル・デバイスに対する既定のニーモニックを指定します。[OPEN](#) コマンドまたは [USE](#) コマンドにニーモニック空間の引数が含まれない場合、InterSystems IRIS はこのデバイス・タイプ用の既定のニーモニックを使用します。詳細は、“[ニーモニック空間によるデバイス制御](#)” を参照してください。

このパラメータの変更

管理ポータル の [\[IO 設定\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[デバイス設定\]](#) > [\[IO 設定\]](#)) で、[\[Terminal\]](#) の行に InterSystems IRIS ルーチン名を入力します。

管理ポータルを使用する代わりに、Config.IO クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、Terminal を変更できます。

[Journal]

ここでは、構成パラメータ・ファイルの [Journal] セクションのパラメータについて説明します。

AlternateDirectory

ジャーナル・ファイルの代替保存場所を設定します。

構文

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

n は、既存のディレクトリのフル・パスです。既定では、ディレクトリは記述されていません。

説明

AlternateDirectory は、ジャーナル・ファイルを格納する代替 (二次) ディレクトリの名前です。

この代替ディレクトリが使用されるのは、[CurrentDirectory](#) で指定されているプライマリ・ジャーナル・ディレクトリが利用できない場合です。例えば、プライマリ・ジャーナル・ディレクトリのディスク・パーティションが容量不足になるか、オフラインになるなどの問題がある場合です。この理由から、代替ジャーナル・ディレクトリは、現在のジャーナル・ディレクトリとは別のディスクに配置することをお勧めします。

インストール時のこのフィールドの初期値は、空の文字列です。InterSystems IRIS® Data Platform を初めて起動したときに、実際のジャーナル・ディレクトリの場所 (<install-dir>%mgr%journal など) がここに入力されます。

AlternateDirectory パラメータの長さと [JournalFilePrefix](#) パラメータの長さを合わせた長さが 208 文字を超えてはいけません (JournalFilePrefix の最大長は 64 文字です)。

このパラメータの変更

管理ポータル の [ジャーナル設定] ページ ([システム管理] > [構成] > [システム構成] > [ジャーナル設定]) で、[二次ジャーナルディレクトリ] の行の [参照] を選択します。既存のディレクトリの名前を選択します。

管理ポータルを使用する代わりに、`JOURNAL` ユーティリティ ("[JRNOPTS を使用したジャーナル設定の更新](#)" を参照) から `Config.Journal` クラスで (クラス・リファレンスを参照)、または CPF をテキスト・エディタで編集することで ("[アクティブな CPF の編集](#)" を参照)、AlternateDirectory を変更できます。

ArchiveName

ジャーナル・ファイルをアーカイブする際のコピー先を指定します。

構文

```
[Journal]      ArchiveName=archivename
```

archivename は、[アーカイブ・ターゲット](#)の名前です。既定値はありません。

説明

ArchiveName は、[Archives] セクションで定義される[アーカイブ・ターゲット](#)の名前です。.

このパラメータの変更

管理ポータルの [ジャーナル設定] ページ ([システム管理]→[構成]→[システム構成]→[ジャーナル設定]) で、[Archive journal files] セクションの [To Archive Target] ドロップダウン・リストから既存のターゲットを選択します。

管理ポータルを使用する代わりに、`^JOURNAL` ユーティリティ ("[^JRNOPTSを使用したジャーナル設定の更新](#)"を参照) から `Config.Journal` クラスで (クラス・リファレンスを参照)、または CPF をテキスト・エディタで編集することで ("[アクティブな CPF の編集](#)"を参照)、ArchiveName を変更できます。

BackupsBeforePurge

完了したジャーナル・ファイルを InterSystems IRIS® Data Platform が削除するまでのバックアップ回数を設定します。

構文

```
[Journal]      BackupsBeforePurge=n
```

n は、0 から 10 の範囲の整数です。既定値は 2 です。

説明

BackupsBeforePurge は、完了したジャーナル・ファイル（つまり、使用されていないジャーナル・ファイル）を削除するタイミングを定義します。値 n は、ジャーナル・ファイルを削除するまでに実行する正常な InterSystems IRIS インスタンス・バックアップの回数です。

BackupsBeforePurge は、[DaysBeforePurge](#) と関連しています。両方に 0 よりも大きい値を指定した場合は、n 日が経過した時点と正常なバックアップの回数が n 回に到達した時点のうち、早い方の時点でファイルが削除されます。BackupsBeforePurge が 0 の場合は、DaysBeforePurge のみを基準として削除され、DaysBeforePurge が 0 の場合は、BackupsBeforePurge のみを基準として削除されます。両方が 0 の場合は、ジャーナル・ファイル（およびジャーナル履歴）の自動削除は行われず、ジャーナル・ファイルは削除されなくなります。

現在開いているトランザクションが含まれているジャーナル・ファイルは、上記の条件を満たしていても削除されません。

注釈 [PurgeArchived](#) が 1 の場合、この設定は無視されます。

ジャーナル・ファイルの詳細は、“[ジャーナリングの概要](#)”を参照してください。

このパラメータの変更

管理ポータル の [ジャーナル設定] ページ ([システム管理] > [構成] > [システム構成] > [ジャーナル設定]) の [ジャーナルファイルを削除するタイミング] カテゴリで、次のいずれかを選択します。

- ・ 日数に基づいてジャーナル・ファイルを削除するには、[この日数後] を選択して日数を入力します。
- ・ バックアップ回数に基づいてジャーナル・ファイルを削除するには、[この回数のバックアップ成功後] を選択してバックアップ回数を入力します。

管理ポータルを使用する代わりに、`^JOURNAL` ユーティリティ (“[JRNOPS を使用したジャーナル設定の更新](#)”を参照) から `Config.Journal` クラスで (クラス・リファレンスを参照)、または CPF をテキスト・エディタで編集することで (“[アクティブな CPF の編集](#)”を参照)、BackupsBeforePurge を変更できます。

CompressFiles

完了したジャーナル・ファイルを圧縮するかどうかを指定します。

構文

```
[Journal]    CompressFiles=n
```

n は 1 または 0 です。既定値は 1 です。

概要

CompressFiles が有効な場合 (n=1)、InterSystems IRIS® データ・プラットフォームは終了したジャーナル・ファイルを自動的に圧縮します。システムは、完了したジャーナル・ファイルを定期的にスキャンし、Zstd 圧縮を使用して圧縮します。アクティブなジャーナル・ファイルは、完了するまで圧縮されません。

圧縮されたジャーナル・ファイルは同じ名前を保持しますが、末尾に **z** が追加されます。例えば、ジャーナル・ファイル **20210818.001** を圧縮すると、**20210818.001z** になります。InterSystems IRIS のほとんどの関数とユーティリティは、**z** は無視して、元の名前で圧縮したジャーナル・ファイルにアクセスできます。

ジャーナル・ファイルの詳細は、“[ジャーナリングの概要](#)” を参照してください。

このパラメータの変更

管理ポータル[の](#) **[ジャーナル設定]** ページ (**[システム管理]** > **[構成]** > **[システム構成]** > **[ジャーナル設定]**) で、**[Compress journal files]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、`^JOURNAL` ユーティリティ (“[^JRNOPS を使用したジャーナル設定の更新](#)” を参照) から `Config.Journal` クラスで (クラス・リファレンスを参照)、または CPF をテキスト・エディタで編集することで (“[アクティブな CPF の編集](#)” を参照)、CompressFiles を変更できます。

CurrentDirectory

ジャーナル・ファイルの主たる保存場所を設定します。

構文

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

n は、既存のディレクトリのフル・パスです。既定は <install-dir>%mgr%journal% です。

説明

CurrentDirectory は、ジャーナル・ファイルを格納するディレクトリ (主ディレクトリ) の名前です。インストール時のこのフィールドの初期値は、空の文字列です。InterSystems IRIS® Data Platform を初めて起動したときに、実際のジャーナル・ディレクトリの場所 (<install-dir>%mgr%journal など) がここに入力されます。

CurrentDirectory パラメータの長さと [JournalFilePrefix](#) パラメータの長さを合わせた長さが 208 文字を超えてはいけません (JournalFilePrefix の最大長は 64 文字です)。

このパラメータの変更

管理ポータル の [\[ジャーナル設定\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[システム構成\]](#) > [\[ジャーナル設定\]](#)) で、[\[主ジャーナルディレクトリ\]](#) の行の [\[参照\]](#) を選択します。既存のディレクトリの名前を選択します。

管理ポータルを使用する代わりに、`^JOURNAL` ユーティリティ ("[JRNOPTS を使用したジャーナル設定の更新](#)" を参照) から `Config.Journal` クラスで ([クラス・リファレンス](#) を参照)、または CPF をテキスト・エディタで編集することで ("[アクティブな CPF の編集](#)" を参照)、CurrentDirectory を変更できます。

DaysBeforePurge

完了したジャーナル・ファイルを InterSystems IRIS® Data Platform が削除するまでの日数を設定します。

構文

```
[Journal]    DaysBeforePurge=n
```

n は、0 から 100 の範囲の整数です。既定値は 2 です。

説明

DaysBeforePurge は、完了したジャーナル・ファイル（つまり、使用されていないジャーナル・ファイル）を削除するタイミングを定義します。値 n は、ジャーナル・ファイルを削除するまでの経過日数です。

BackupsBeforePurge は、[DaysBeforePurge](#) と関連しています。両方に 0 よりも大きい値を指定した場合は、n 日が経過した時点と正常なバックアップの回数が n 回に到達した時点のうち、早い方の時点でファイルが削除されます。BackupsBeforePurge が 0 の場合は、DaysBeforePurge のみを基準として削除され、DaysBeforePurge が 0 の場合は、BackupsBeforePurge のみを基準として削除されます。両方が 0 の場合は、ジャーナル・ファイル（およびジャーナル履歴）の自動削除は行われず、ジャーナル・ファイルは削除されなくなります。

現在開いているトランザクションが含まれているジャーナル・ファイルは、上記の条件を満たしていても削除されません。

注釈 [PurgeArchived](#) が 1 の場合、この設定は無視されます。

ジャーナル・ファイルの詳細は、“[ジャーナリングの概要](#)” を参照してください。

このパラメータの変更

管理ポータルでの [ジャーナル設定] ページ ([システム管理] > [構成] > [システム構成] > [ジャーナル設定]) の [ジャーナルファイルを削除するタイミング] カテゴリで、次のいずれかを選択します。

- ・ 日数に基づいてジャーナル・ファイルを削除するには、[この日数後] を選択して日数を入力します。
- ・ バックアップ回数に基づいてジャーナル・ファイルを削除するには、[この回数のバックアップ成功後] を選択してバックアップ回数を入力します。

管理ポータルを使用する代わりに、`^JOURNAL` ユーティリティ (“[JRNOPTS を使用したジャーナル設定の更新](#)” を参照) から `Config.Journal` クラスで (クラス・リファレンスを参照)、または CPF をテキスト・エディタで編集することで (“[アクティブな CPF の編集](#)” を参照)、DaysBeforePurge を変更できます。

FileSizeLimit

ジャーナル・ファイルの最大サイズを設定します。

構文

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

n は、0 から 4,079 (MB) の範囲の整数です。既定値は 1024 です。

説明

FileSizeLimit は、ジャーナル・ファイルの最大サイズ (MB) です。ジャーナル・ファイルのサイズが大きくなり、このサイズに達すると、そのファイルは閉じられ、新しいジャーナル・ファイルが作成されます。ジャーナル・ファイルのロールオーバーの詳細は、“[ジャーナル・ファイル名とロールオーバー](#)” を参照してください。

FileSizeLimit パラメータと [jrnbufs](#) パラメータは、これの合計サイズの最大値が 4 GB に制限されています。詳細は、“[ジャーナル設定の構成](#)” を参照してください。

このパラメータの変更

管理ポータルの [ジャーナル設定] ページ ([システム管理] > [構成] > [システム構成] > [ジャーナル設定]) で、[次のサイズに達すると新規ジャーナルに切り替える] の行にメガバイト単位の数値を入力します。

管理ポータルを使用する代わりに、`^JOURNAL` ユーティリティ (“[JRNOPS を使用したジャーナル設定の更新](#)” を参照) から `Config.Journal` クラスで (クラス・リファレンスを参照)、または CPF をテキスト・エディタで編集することで (“[アクティブな CPF の編集](#)” を参照)、FileSizeLimit を変更できます。

FreezeOnError

ジャーナル入出力エラーの発生時にジャーナリングの凍結を許可します。

構文

```
[Journal] FreezeOnError=n
```

n は 1 または 0 です。既定値は 0 です。

説明

FreezeOnError=0 (偽、既定) の場合、ジャーナル・ファイルの入出力エラー発生時にジャーナリングは凍結されません。このオプションによってインスタンスの可用性が確保されますが、データ損失が生じる可能性があります。

ジャーナル・デーモンは、失敗した操作を定期的 (通常は 1 秒おき) に再試行します。これは、成功するか、あるいはインスタンスがジャーナリングされた更新をこれ以上バッファできなくなるか、事前に設定されている時間制限 (通常は 150 秒) に達したために、ジャーナリングが無効になるまで続きます。

重要 ジャーナリングが無効になった場合は、直ちにデータベースのバックアップを行ってください。ジャーナリングなしで続行するリスクを考慮する必要があります。エラーが発生した期間中の動作はリストアされません。

無効になったジャーナリングは、手動で再開する必要があります。それを行うには、[JRNSTART](#) ルーチンを実行するか、[JOURNAL](#) ルーチン・メニューからオプション 1 [] を選択します。

FreezeOnError=1 (真) の場合、ジャーナル・ファイルの入出力エラーが発生すると、ジャーナリングされたグローバル更新はすべて即座に凍結されます。ジャーナル・デーモンがジャーナル書き込みを最低 30 秒間完了できない場合にも、グローバル更新は凍結されます。このオプションによってインスタンスはデータ損失から守られますが、問題を解決している間、インスタンスは可用性が低下するか、使用できなくなります。

ジャーナル・デーモンは失敗した入出力操作を再試行し、成功するとグローバル更新の凍結を解除します。グローバル更新がフリーズしている間、その他のジョブも停止されます。一般的な結果として、ジャーナリングの問題が解決するまで InterSystems IRIS は停止したままとなり、エンドユーザにはシステムの運用が停止したように見えます。InterSystems IRIS が停止している間に問題の対策をとることができます。例えば、ディスクの領域の解放、別のディスクへのジャーナル切り替え、ハードウェア障害の修正などの措置をとります。

重要 FreezeOnError の設定は自動的にオーバーライドされ、インスタンスが[ミラーのフェイルオーバー・メンバ](#)である場合にオンになります。

詳細は、“[ジャーナル入出力エラー](#)” を参照してください。

このパラメータの変更

管理ポータル の [\[ジャーナル設定\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[システム構成\]](#) > [\[ジャーナル設定\]](#)) で、[\[FreezeOnError\]](#) を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Journal クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、FreezeOnError を変更できます。

JournalFilePrefix

ジャーナル・ファイル名の接頭語を設定します。

構文

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

n は英数字文字列です。最大長は 64 文字です。既定値は空文字列です。

説明

JournalFilePrefix は、ジャーナル・ファイル名の先頭に追加される文字列です。例えば、JournalFilePrefix を Oct に設定すると、結果のジャーナル・ファイルは **Oct20221001.001** のようになります。

このパラメータの変更

管理ポータル の [ジャーナル設定] ページ ([システム管理] > [構成] > [システム構成] > [ジャーナル設定]) で、[ジャーナルファイル接頭子] の行に接頭語を入力します。

管理ポータルを使用する代わりに、`^JOURNAL` ユーティリティ ("[JRNOPS を使用したジャーナル設定の更新](#)" を参照) から `Config.Journal` クラスで (クラス・リファレンスを参照)、または CPF をテキスト・エディタで編集することで ("[アクティブな CPF の編集](#)" を参照)、`JournalFilePrefix` を変更できます。

JournalcspSession

Web セッションのジャーナリングを許可します。

構文

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

n は 1 または 0 です。既定値は 0 です。

説明

JournalcspSession が有効な場合 (n = 1)、InterSystems IRIS® データ・プラットフォームは、`%cspSession` グローバルをジャーナルします。フェイルオーバーを目的として Web セッション・グローバルを別のマシン上に複製する場合や InterSystems IRIS を再起動しても Web セッションが保持されるようにする場合に、この設定を有効にします。この設定を有効にしない場合、`%cspSession` グローバルは **IRISTEMP** にマッピングされ、ジャーナルされません。システムの再起動時、または新しい InterSystems IRIS ソフトウェア・バージョンへのアップグレード時に、`%cspSession` グローバルが削除され、それまでの Web セッションのレコードは削除されます。

このパラメータの変更

管理ポータル の **[ジャーナル設定]** ページ (**[システム管理]** > **[構成]** > **[システム構成]** > **[ジャーナル設定]**) で、**[ウェブセッションをジャーナルする]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Journal クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、JournalcspSession を変更できます。

PurgeArchive

ジャーナル・ファイルがアーカイブ・ターゲットにコピーされたらすぐにそれらのファイルをパージするかどうかを指定します。

構文

```
[Journal]    PurgeArchive=boolean
```

boolean は 1 または 0 です。既定値は 0 です。

説明

PurgeArchive は、ジャーナル・ファイルがアーカイブ・ターゲットにコピーされたらすぐにそれらのファイルをパージするかどうかを指定します。この設定に関係なく、必要なファイルがパージされることはありません。

PurgeArchive が 1 の場合、[BackupsBeforePurge](#) および [DaysBeforePurge](#) の設定は共に無視されます。

このパラメータの変更

管理ポータルの [ジャーナル設定] ページ ([システム管理]→[構成]→[システム構成]→[ジャーナル設定]) で、[Archive journal files] セクションの [As soon as they are copied to archive] チェック・ボックスにチェックを付けます (このオプションを表示するには、アーカイブ・ターゲットを選択する必要があります)。

管理ポータルを使用する代わりに、`^JOURNAL` ユーティリティ ("[JRNOPTS を使用したジャーナル設定の更新](#)" を参照) から `Config.Journal` クラスで (クラス・リファレンスを参照)、または CPF をテキスト・エディタで編集することで ("[アクティブな CPF の編集](#)" を参照)、PurgeArchive を変更できます。

[LicenseServers]

このトピックでは、CPF の [LicenseServers] セクションの LicenseServer パラメータについて説明します。複数のライセンス・サーバを定義できます。

LicenseServers

ライセンス・サーバを定義します。

構文

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

説明

[LicenseServers] セクションには、InterSystems IRIS® Data Platform 用に構成されているすべてのライセンス・サーバのエントリが含まれます。各エントリは以下のとおりです。

- ・ IpAddress – ライセンス・サーバの IP アドレス。
- ・ Port – ライセンス・サーバが使用する UDP ポート番号。異なる IP アドレスで使用されているポート番号は、異なる必要はありません。ただし、各 IP アドレスで使用されているライセンス・サーバのポート番号は、その IP アドレスで使用されている UDP ポート番号とは異なる必要があります。
- ・ KeyDirectory – ライセンス・キーのディレクトリ。この設定はオプションです。ディレクトリを指定すると、起動時にライセンス・サーバはそのディレクトリから有効な .key ファイルをすべて読み取ります。

ライセンス・サーバの詳細は、“[InterSystems IRIS ライセンスの構成](#)”を参照してください。

このパラメータの変更

管理ポータルの [ライセンスサーバ] ページ ([システム管理] > [ライセンス] > [ライセンスサーバ]) で、編集するライセンス・サーバの名前を選択するか、[ライセンス・サーバを作成] をクリックして新しいライセンス・サーバを追加します。

[Logging]

このトピックでは、CPF の [Logging] セクションのパラメータについて説明します。

ChildProcessLaunchCommand

システムからの構造化ログ・データの送信先を指定するパイプ・コマンドを設定します。

構文

```
[Logging] ChildProcessLaunchCommand=Executable Options
```

既定の形式は、以下のとおりです。

```
ChildProcessLaunchCommand=irislogd.exe -f /path/to/myfilename.log
```

説明

ChildProcessLaunchCommand は、システムからの構造化ログ・データの送信先を指定します。これは、ログ・データを受信する、ターゲットの実行可能ファイルを取ります。必要に応じて、その実行可能ファイルに引数を指定できます。

既定の `irislogd` 実行可能ファイルの場合、以下の引数を渡すことができます。

引数	目的
-d	診断メッセージとエラー・メッセージを発行します。
-e errfilename	指定されたファイルにエラー・メッセージと診断メッセージを書き込みます。
-f logfilename	指定されたファイルにログ・メッセージを書き込みます。
-h hostname	指定されたホスト名を構造化ログ・ファイルに含めます。
-i irisinstance	指定されたインスタンス名を構造化ログ・ファイルに含めます。
-s	ログ・メッセージを Unix® syslog ファシリティに書き込みます (Unix® のみ)。

このパラメータの変更

このパラメータは、管理ポータル、LOGDMN ルーチン、またはクラスベースの API (すべて "[Configure Structured Logging](#)" を参照) を使用して変更できます。または、テキスト・エディタで CPF を編集することもできます ("[アクティブな CPF の編集](#)" を参照)。

Enabled

構造化ログ・データの収集を許可します。

構文

```
[Logging] Enabled=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

構造化ログを有効にすると (Enabled=1)、InterSystems IRIS® データ・プラットフォームの起動時に、ログ・デーモンが自動的に有効になります。ただし、これを 1 に設定しても、ログ・デーモンは起動しません。このパラメータを 0 に設定すると、ログ・デーモンを開始しようとしてもすべて失敗します。

このパラメータの変更

このパラメータは、管理ポータル、`^LOGDMN` ルーチン、またはクラスベースの API (すべて ["Configure Structured Logging"](#) を参照) を使用して変更できます。または、テキスト・エディタで CPF を編集することもできます (["アクティブな CPF の編集"](#) を参照)。

Format

パイプに送信するデータの形式。

構文

```
[Logging] Format=n
```

n は、名前と値のペア (NVP) または JSON です。既定値は NVP です。

説明

ログ・デーモンは、構造化ログの出力形式を NVP または JSON に設定できます。NVP 形式では、ファイル内の各行に、スペースで区切られた一連の名前と値のペアが含まれます。名前と値の各ペアの形式は、name=value で、value にスペース文字が含まれる場合は、value が括弧で囲まれます。JSON 形式では、ファイル内の各行が、一連のプロパティを持つ JSON オブジェクトです。

このパラメータの変更

このパラメータは、管理ポータル、`LOGDMN` ルーチン、またはクラスベースの API (すべて ["Configure Structured Logging"](#) を参照) を使用して変更できます。または、テキスト・エディタで CPF を編集することもできます (["アクティブな CPF の編集"](#) を参照)。

interval

パイプ・コマンドへの連続した呼び出しの間隔 (秒数) を設定します。

構文

```
[Logging] Interval=n
```

n は、0.01 から 3600 の範囲の数値です。既定値は 10 です。

説明

Interval は、ログ・デーモンが **messages.log** ファイルおよび **%SYS.Audit** ログに対して新規エントリがないかスキャンするまでの待機時間を秒数で表します。小数点以下最大 2 桁の数値で表すことができ、最小 0.01 (10 ミリ秒) から最大 3600 (1 時間) の範囲の数値に指定できます。

このパラメータの変更

このパラメータは、管理ポータル、`LOGDMN` ルーチン、またはクラスベースの API (すべて ["Configure Structured Logging"](#) を参照) を使用して変更できます。または、テキスト・エディタで CPF を編集することもできます (["アクティブな CPF の編集"](#) を参照)。

Level

最小ログ・レベルを設定します。

構文

```
[Logging] Level=n
```

n は、以下のいずれかです。

- ・ DEBUG2
- ・ DEBUG
- ・ INFO
- ・ WARN
- ・ SEVERE
- ・ FATAL

既定値は WARN です。

説明

このパラメータは、以下のように、ログの最小詳細レベルを表します。

- ・ DEBUG2 – 詳細なデバッグ・メッセージ (16 進ダンプなど)。
- ・ DEBUG – それほど詳細ではないデバッグ・メッセージ。
- ・ INFO – すべての監査イベントを含む情報メッセージ。
- ・ WARN (既定値) – 警告。注意が必要な可能性はあるが、処理の中断は生じていない問題を示します。
- ・ SEVERE – 重大なエラー。処理の中断を招いた問題が発生したことを示します。
- ・ FATAL – 致命的なエラー。この問題によってシステムが動作しなくなったことを示します。

このパラメータの変更

このパラメータは、管理ポータル、LOGDMN ルーチン、またはクラスベースの API (すべて "[Configure Structured Logging](#)" を参照) を使用して変更できます。または、テキスト・エディタで CPF を編集することもできます ("[アクティブな CPF の編集](#)" を参照)。

[Map]

このトピックでは、CPF の [Map] セクションのパラメータについて説明します。[Map.w] というセクションに存在するパラメータについても説明します。w は、InterSystems IRIS® Data Platform ネームスペースの名前です。

Global

グローバルをネームスペースにマッピングします。

構文

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

説明

ネームスペース (w) にマッピングが含まれている場合、InterSystems IRIS® Data Platform は、そのネームスペースに対応する [Map.w] という名前のセクションを構成パラメータ・ファイル (CPF) 内に作成します。例えば、USER ネームスペースのマッピングは、セクション [Map.USER] に記述されます。[Map.w] セクションには、そのネームスペースのグローバル・マッピング、ルーチン・マッピング、およびパッケージ・マッピングがすべて含まれます。

グローバル・マッピングは Global_GlobalName の形式をとります。GlobalName はマッピングされる特定のグローバルです。各グローバル・エントリには、ネームスペースにグローバルをマッピングする値がコンマで区切って記述されています。最初の値のみが必須です。その他の値は、値を指定しない場合、インスタンスの既定値に設定されます。これらの値は以下のとおりです。

- Database — グローバルのデータベースの場所。
- Collation — グローバルの照合 (既定値は 5。これは InterSystems IRIS 標準の照合です)。
- LockLocation — グローバルのロック・データベースの場所 (既定値は Database の場所)。ロック・データベースの場所は、Database の場所と同じにすることをお勧めします。

グローバルが添え字レベルでマッピングされていない場合、照合設定は無視されます。実際のグローバルの照合が、ネームスペース定義で定義された照合と (グローバルにマッピングされた添え字レベルで) 一致しない場合、参照すると <COLLATEMISMATCH> エラーが発生します。これは、InterSystems IRIS では、特定のデータベースの既定の照合に関係なく、グローバルが存在しているすべてのデータベースで同じ照合を持つ必要があるためです。

マッピングの詳細は、"[ネームスペースへのグローバル、ルーチン、およびパッケージ・マッピングの追加](#)" を参照してください。

添え字マッピング

GlobalName の一部として添え字マッピングを指定できます (ss)。添え字マッピングを指定する場合は、グローバル自体の上位マッピングが存在する必要があります。つまり、マッピング `X(9)` を作成する場合は、`X` に対するマッピングが存在する必要があります。添え字マッピングは以下の形式をとることができます。

(1)

("A")

(1):(5) — 1 ~ 4 (5 は含まれません)

("A"):("Z") — A ~ Y (Z は含まれません)

(BEGIN):("X") — 先頭 ~ X (X 自体は含まれません)

("Y"):(END) — Y ~ 末尾

例

以下では、グローバル `^SALES` を SALES データベースにマッピングします。

```
Global_SALES=SALES
```


以下では、グローバル `SALES("MA")` を SALESMA データベースに添え字マッピングします。

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

以下では、グローバル `ACCOUNT` を ACCOUNTS データベースにマッピングします。

```
Global_ACCOUNT=ACCOUNTS
```

以下では、グローバル `ACCOUNT(1) ~ `ACCOUNT(4)` を ACCOUNTS1TO4 データベースに添え字マッピングします(`ACCOUNT(5)` は含まれません)。

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

以下では、先頭が ABC である名前を持つすべてのグローバルを ABC データベースにマッピングします。

```
Global_ABC*=ABC
```

このパラメータの変更

管理ポータルの [ネームスペース] ページ ([システム管理] > [構成] > [システム構成] > [ネームスペース]) で、対象とするネームスペースの [グローバルマッピング] を選択します。新しいエントリを追加するには、[新規作成] をクリックします。既存のエントリを編集するには、そのエントリの行の [編集] を選択します。

Package

パッケージをネームスペースにマッピングします。

構文

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

説明

ネームスペース (w) にマッピングが含まれている場合、InterSystems IRIS® Data Platform は、そのネームスペースに対応する [Map.w] という名前のセクションを構成パラメータ・ファイル (CPF) 内に作成します。例えば、USER ネームスペースのマッピングは、セクション [Map.USER] に記述されます。[Map.w] セクションには、そのネームスペースのグローバル・マッピング、ルーチン・マッピング、およびパッケージ・マッピングがすべて含まれます。

パッケージ・マッピングは Package_PackageName の形式をとります。PackageName はマッピングされる特定のパッケージです。各パッケージ・エントリには、指定したパッケージが含まれるデータベースの場所 (Database) が含まれます。指定されたパッケージ内のクラスは、w ネームスペースで使用可能になります。

マッピングの詳細は、[“ネームスペースへのグローバル、ルーチン、およびパッケージ・マッピングの追加”](#) を参照してください。

例

パッケージ TEST を USER データベースにマッピングします。

```
Package_TEST=USER
```

パッケージ TOOLS を DEVELOPER データベースにマッピングします。

```
Package_TOOLS=DEVELOPER
```

このパラメータの変更

管理ポータルの [ネームスペース] ページ ([システム管理] > [構成] > [システム構成] > [ネームスペース]) で、対象とするネームスペースの [パッケージマッピング] を選択します。新しいエントリを追加するには、[新規作成] をクリックします。既存のエントリを編集するには、そのエントリの行の [編集] を選択します。

Routine

ルーチンをネームスペースにマッピングします。

構文

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

説明

ネームスペース (w) にマッピングが含まれている場合、InterSystems IRIS® Data Platform は、そのネームスペースに対応する [Map.w] という名前のセクションを構成パラメータ・ファイル (CPF) 内に作成します。例えば、**USER** ネームスペースのマッピングは、セクション [Map.USER] に記述されます。[Map.w] セクションには、そのネームスペースのグローバル・マッピング、ルーチン・マッピング、およびパッケージ・マッピングがすべて含まれます。

ルーチン・マッピングは Global_RoutineName の形式をとります。RoutineName はマッピングされる特定のルーチンです。各ルーチン・エントリには、指定したルーチンが含まれるデータベースの場所 (Database) が含まれます。指定されたルーチンは、w ネームスペースで使用可能になります。

Type は通常指定しません。Type の指定が必要となるのは、ルーチンの一部を別のデータベースにマッピングする場合のみです。Type の有効な値は、MAC、INT、INC、または OBJ のいずれかです。

マッピングの詳細は、"[ネームスペースへのグローバル、ルーチン、およびパッケージ・マッピングの追加](#)" を参照してください。

例

以下では、ルーチン SALE を SALES データベースにマッピングします。

```
Routine_SALE=SALES
```

以下では、先頭が ACC である名前を持つすべてのルーチンを ACCOUNTS データベースにマッピングします。

```
Routine_ACC*=ACCOUNTS
```

以下では、ルーチン TEST のオブジェクト・コードを TEST データベースにマッピングします。

```
Routine_TEST_OBJ=TEST
```

このパラメータの変更

管理ポータルの [ネームスペース] ページ ([システム管理] > [構成] > [システム構成] > [ネームスペース]) で、対象とするネームスペースの [ルーチンマッピング] を選択します。新しいエントリを追加するには、[新規作成] をクリックします。既存のエントリを編集するには、そのエントリの行の [編集] を選択します。

[MapMirrors]

このトピックでは、CPF の [MapMirrors] セクションについて説明します。

MapMirrors

このインスタンスを含むミラーのミラー・メンバを定義します。

構文

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

説明

現在のインスタンスに接続している各ミラー (m) に対して、InterSystems IRIS® Data Platform は、CPF に [MapMirrors.m] というセクションを作成します。このセクションには、そのミラーに接続しているすべてのインスタンスのエントリが含まれます。以下に示すように、エントリは Name と 14 個のコンマ区切り値 (a ~ n) で構成されます。

- ・ Name - 必須。このメンバをミラー内で識別する一意の名前。最大 15 文字の大文字の英数字の文字列には、スペースやタブ、コンマ (,), セミコロン (;), および等号 (=) を含めることはできません。また、この文字列は、保存前に大文字に変換されます。

これは、ミラー内のこのインスタンスの名前です。ミラー・システム名は、すべてのミラーにわたり一意である必要があります。インスタンスは、複数のミラー・セットに出現する可能性のある単一のミラー・システム名を持つからです (つまり、非同期メンバは複数のミラーに接続する可能性があります)。ほとんどの場合は、GUID を使用してミラー・メンバが識別され、Name は表示のために使用されます。この名前にコロン (:) を含めることはできません。
- ・ a (AgentAddress) - プライマリに接続するミラー・メンバが、このフェイルオーバー・メンバ上のエージェントと通信するために使用する必要があるネットワーク・アドレス (DNS の問題を回避するために推奨される IP アドレス)。これは、非同期メンバ上では省略されます。エージェントは、これらのインスタンス上のミラーリングには使用されないからです。これは、フェイルオーバー・メンバ上では必須です。エージェントはジャーナル・データを転送できるため、ネットワーク輻輳を避けるためにここではプライベート・アドレスであることが望ましい場合があります。
- ・ b (AgentPort) - このインスタンス上のエージェントがリッスンするように構成されているポート番号。ローカル・エージェントがアクティブの場合は、`##class(SYS.Agent).GetApplicationPort()` は現在の値を返します。
- ・ c - 内部用。
- ・ d (SuperServerAddress) - 外部ミラー認識システムがプライマリに接続するために使用するネットワーク・アドレス (これは、現時点では ECP アプリケーション・サーバのみですが、将来は別の接続にも拡張される可能性があります)。他のミラー・メンバが、制御や監視を目的としてメンバのスーパーサーバ・アドレスに接続する場合があります。メンバがプライマリであり、プライマリのミラー・プライベート・アドレス (MirrorPrivate) にアクセスできない場合、非同期メンバはこのアドレスを使用して (ジャーナル・データを受け取るための) データ・チャンネルを確立しようとします。
- ・ e (GUID) - 必須項目。このミラーで一意の内部 GUID。ミラー内のこのノードを一意に識別します。ノードの識別以外の主な用途は、ミラーリングされたデータベースの特定コピーを所有するインスタンスを識別することです。
- ・ f (InstanceDirectory) - インスタンスのインストール・ディレクトリ (mgr ディレクトリの親ディレクトリ)。エージェントに対してインスタンスを識別するために、主にフェイルオーバー・メンバ上で使用されます。
- ・ g (MemberType) - ミラー・メンバのタイプを示す数値。次のいずれかです。
 - 0 - フェイルオーバー・メンバ
 - 2 - 非同期メンバ
- ・ h (MirrorPrivate) - このインスタンスがプライマリの場合、他のミラー・メンバはこのアドレスを使用して、プライマリからジャーナル・データを受け取るためのミラー・データ・チャンネルを確立します。ミラー・プライベート・アドレスでこれにアクセスできない場合、非同期メンバは、プライマリのスーパーサーバ・アドレス (SuperServerAddress) を代わりに利用します。

- ・ i (MirrorSSPort) - このインスタンス用のスーパーサーバ・ポート。このインスタンスに対する接続を確立するクライアントによって、MirrorAddress および ECPAddress の両方と共に使用されます。
- ・ j、k、l、m、および n - 内部用。

例

各エントリは次のように 1 行で記述します。

```
[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,
```

このパラメータの変更

管理ポータル の [ミラーの作成] ページ ([システム管理] > [構成] > [ミラー設定] > [ミラーの作成]) で、要求された情報を入力します。[ミラーの編集] ページで既存のミラーを編集できますが、編集操作を実行できるのはプライマリ・フェイルオーバー・メンバからのみです。

[MirrorMember]

このトピックでは、CPF の [MirrorMember] セクションのパラメータについて説明します。

AgentAddress

未使用。

説明

未使用。

AsyncMemberGUID

非同期メンバ GUID を確認します。

構文

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

Name は英数字文字列です。

説明

非同期メンバと呼ばれるミラー・メンバを作成できます。これは、企業全体の1つ以上のミラーから更新を受信するように構成できます。これにより、単一のノードが包括的な全社規模のデータ・ウェアハウスとして機能できるようになります。非同期メンバは、ミラーに属していません。したがって、フェイルオーバーの候補ではありません。

詳細は、“[非同期ミラー・メンバ](#)”を参照してください。

例

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

このパラメータの変更

Config.MirrorMember クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)”を参照)、AsyncMemberGUID を変更できます。

AsyncMemberType

非同期タイプ (災害復旧、読み取り専用、または読み取り/書き込み) を指定します。

構文

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

n は、0、1、または 2 です。

説明

AsyncMemberType は、非同期メンバが災害復旧 (DR)、読み取り専用レポート、または読み取り/書き込みレポートのいずれの非同期メンバであるかを示します。

- ・ 0 - 災害復旧 (DR)。これは災害復旧の非同期メンバです。このメンバのミラーリングされたデータベースはすべて、読み取り専用のミラーリングされたデータベースです。
- ・ 1 - 読み取り専用レポート。これはレポート非同期メンバです。このメンバのミラーリングされたデータベースはすべて、読み取り専用または読み取り/書き込みのデータベースです。データベースの作成時は、既定で読み取り専用です。
- ・ 2 - 読み取り/書き込みレポート。これはレポート非同期メンバです。このメンバのミラーリングされたデータベースはすべて、読み取り専用または読み取り/書き込みのデータベースです。データベースの作成時は、既定で読み取り/書き込みです。

詳細は、“[非同期ミラー・メンバ](#)” を参照してください。

このパラメータの変更

管理ポータルの [非同期として参加] ページ ([システム管理] > [構成] > [ミラー設定] > [非同期として参加]) で、[ミラー情報] に情報を入力して [次へ] を選択します。[非同期メンバ情報] ページの [非同期メンバのシステム・タイプ] の行で、ドロップダウン・リストからタイプを選択します。

管理ポータルを使用する代わりに、Config.MirrorMember クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、AsyncMemberType を変更できます。

AsyncUseSystemPurgeInterval

レポート非同期メンバでミラー・ジャーナル・ファイルがどのように削除されるかを指定します。

構文

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

n は 1 または 0 です。

説明

AsyncMemberType は、レポート非同期メンバが、主フェイルオーバー・メンバから受け取ったミラー・ジャーナル・ファイルをどのように削除するかを示します。

- ・ 0 - ミラー・ジャーナル・ファイルは、デジャーナリングされると即時に削除されます。
- ・ 1 - ミラー・ジャーナル・ファイルは、インスタンス・ジャーナル・ファイルの削除条件に従って削除されます。

詳細は、“[非同期メンバの編集または削除](#)”を参照してください。

このパラメータの変更

管理ポータルの **[非同期を編集]** ページ ([システム管理] > [構成] > [ミラー設定] > [非同期を編集]) で、レポート非同期メンバに対して [ミラー・ジャーナル・ファイルの保持] ドロップダウンを使用して、ミラー・ジャーナル・ファイルを削除する方法を決定します。

管理ポータルを使用する代わりに、Config.MirrorMember クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)”を参照)、AsyncUseSystemPurgeInterval を変更できます。

JoinMirror

インスタンスが起動時にミラー構成を処理するかどうかを指定します。

構文

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

n は 1 または 0 です。

説明

JoinMirror が有効な場合 (n = 1)、ミラー構成は処理され、インスタンスは構成に従ってミラー・メンバであると見なされます。

このパラメータが有効ではない場合、ミラー構成は無視され、インスタンスはミラー・メンバとして初期化されません。構成内の問題が原因でインスタンスを起動できない場合や、メンバがミラーへの参加前に再構成される必要がある場合に使用することをお勧めします。例えば、あるインスタンスが以前にプライマリであったが現在はプライマリではない場合、この値を使用すると、そのインスタンスが再起動時にミラーに参加して再びプライマリになろうとすることが防止されます (このインスタンスが再びプライマリになると二重プライマリの状況になります)。

このパラメータの変更

Config.MirrorMember クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、JoinMirror を変更できます。

SystemName

ミラー・メンバの名前を設定します。

構文

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

n は、最大 32 文字の文字列です。

説明

SystemName は、このインスタンスで構成しているフェイルオーバー・メンバの名前です。既定値は、システムのホスト名と InterSystems IRIS® Data Platform インスタンス名を組み合わせた名前です。

ミラー・メンバ名は、保存前に大文字に変換されます。スペース、タブ、または以下の文字は使用できません。

: [] # ; / * = ^ ~ ,

このパラメータの変更

管理ポータル の [ミラーの作成] ページ、[フェイルオーバーとして参加] ページ、または [非同期として参加] ページのいずれか ([システム管理] > [構成] > [ミラー設定]) で、[ミラー・メンバ名] に名前を入力します。

管理ポータルを使用する代わりに、Config.MirrorMember クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、SystemName を変更できます。

ValidatedMember

インスタンスが前のロールでミラーに参加するか、またはミラーに参加する前に現在のプライマリから新しいロールを取得するかを指定します。

構文

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

n は 1 または 0 です。

説明

ValidatedMember が有効な場合 (n = 1)、インスタンスは現在のロールでミラーに参加します。

このパラメータが有効ではない場合、インスタンスは、プライマリに接続して現在のロールを取得してからミラーに参加します。これは、インスタンスおよび ISCAgent が停止しているかアクセスできない間にミラー内でロールの変更があった場合に使用します。例えば、前のバックアップが停止している間に DR 非同期がバックアップを促された場合、ミラーの再始動前に前のバックアップがプライマリから DR 非同期の新しいロールを確実に受け取れるように、インスタンスの再始動前に ValidatedMember を 0 に設定します。

このパラメータの変更

Config.MirrorMember クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、ValidatedMember を変更できます。

VirtualAddressInterface

仮想 IP アドレスを構成する場合、これはその IP アドレスをホストするローカル・ネットワーク・インタフェースの名前です。

構文

```
[MirrorMember]    VirtualAddressInterface=network-interface-name
```

network-interface-name は、仮想 IP アドレスがホストされている現在のプライマリ上のローカル・ネットワーク・インタフェースの識別子です。具体的な形式については、オペレーティング・システムのドキュメントを参照してください。

説明

ミラー仮想 IP (VIP) アドレスを使用すると、すべての外部クライアント (言語バインディング、ODBC/JDBC/SQL クライアントなど) が単一アドレスを使用してミラーに接続できます。詳細は、“[\[Mirrors\]](#)” セクションおよび “[ミラー仮想 IP \(VIP\) の構成](#)” を参照してください。

このパラメータの変更

管理ポータル の [\[ミラーの作成\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[ミラー設定\]](#) > [\[ミラーの作成\]](#)) で、[\[仮想 IP を使用\]](#) に入力します。続いて、[\[IP アドレス\]](#) および [\[マスク \(CIDR 形式\)\]](#) に入力して、[\[ネットワーク・インタフェース\]](#) でネットワーク・インタフェースを選択します。

管理ポータルを使用する代わりに、Config.MirrorMember クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、VirtualAddressInterface を変更できます。

[Mirrors]

このトピックでは、CPF の [Mirrors] セクションについて説明します。

Mirrors

このインスタンスを含むミラーを定義します。

構文

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

説明

[Mirrors] セクションには、InterSystems IRIS® Data Platform の現在のインスタンスに接続されている各ミラーのエントリが含まれます。以下に示すように、エントリは Name と 14 個のコンマ区切り値 (a ~ n) で構成されます。

- ・ Name - 必須。このミラーを識別するときに使用できる一意の名前。最大 15 文字の大文字英数字文字列です。
- ・ a (GUID) - 必須項目。このミラーで一意の内部 GUID。
- ・ b (2) - 内部用。
- ・ c (1) - 内部用。
- ・ d (定義済みのプライマリ) - 一般に保守目的でミラー・フェイルオーバーを無効にする場合に使用されます。InterSystems IRIS または ^MIRROR ルーチンのシャットダウン時に nofailover オプションを使用して操作します。これには、プライマリにする必要があるメンバのミラー名が含まれています。そのノードが起動してプライマリになると自動的にクリアされます。
- ・ e (QOSTimeout) - サービス品質タイムアウト:フェイルオーバー・メンバが、アクションを実行する前に、他のフェイルオーバー・メンバからの応答を待機する最大時間 (ミリ秒単位)。さらに、フェイルオーバー・メンバの応答をアービターが待機する場合にも適用されます。既定は 8000ms です。一般に、専用のローカル・ネットワークに接続された物理 (仮想化されていない) ホストへの導入では、停止に対するより迅速な対応が必要な場合には、この設定を短くすることができます。QoS タイムアウト設定の詳細は、“[サービス品質 \(QoS\) タイムアウトの構成](#)”を参照してください。
- ・ f (0) - 内部用。
- ・ g (UseSSL) - ミラー内でセキュリティを実現するために、SSL/TLS を使用するようにミラーのノードを構成できます。こうすると、1 つのノードから別のノードへの認証と、ノード間での暗号化された通信の両方が提供されます。ミラーで SSL/TLS を使用するために、(フェイルオーバーまたは非同期の) 各メンバは、%Mirror_Client および %Mirror_Server の SSL/TLS 構成のペアを使用します。これらの構成は、ミラーで SSL/TLS が有効になっているときに、各メンバ上に既に存在している必要があります。SSL の設定方法の詳細は、“[ミラー用 TLS 構成の作成および編集](#)”を参照してください。値は 0 (いいえ : 既定値) または 1 (はい) です。
- ・ h (VirtualAddress) - 仮想 IP アドレスを指定します。すべての外部クライアント (言語バインディング、ODBC/JDBC/SQL クライアントなど) が単一アドレスを使用してミラーに接続するように、ミラー仮想 IP (VIP) アドレスを構成することができます。この仮想 IP アドレスは、自動的に現在のプライマリ・メンバ上のインタフェースに結合されます。VIP を使用するには、両方のフェイルオーバー・メンバが同一のサブネットに配置されている必要があります。詳細は、“[ミラー仮想 IP \(VIP\) の構成](#)”を参照してください。
- ・ i (0) - 内部用。
- ・ j (ArbiterNode) - このミラーに対して構成されているアービターのネットワーク・アドレス。アービターは、ミラーのフェイルオーバー・メンバとの継続的な通信が維持される ISCAgent をホストする独立インスタンスで、直接通信できない場合にフェイルオーバーの決定を安全に下すために必要なコンテキストを提供します。
- ・ k (ArbiterPort) - 構成済みアービターの ISCAgent プロセスにより使用されるポート (既定では 2188)。j と同じスペースに縦棒で区切って記述します。

- ・ l (CompressionForFailoverMembers) - ジャーナル・データを主ジャーナルからバックアップに転送する前に圧縮するかどうかを判断します。指定可能な値は、0 (システムによる選択、フェイルオーバー・メンバ間の応答時間が最適化されます)、1 (圧縮しない)、および 2 (圧縮する) です。
- ・ m (CompressionForAsyncMembers) - ジャーナル・データを主ジャーナルから非同期メンバに転送する前に圧縮するかどうかを決定します。指定可能な値は、0 (システムによる選択、ネットワーク使用率が最適化されます)、1 (圧縮しない)、および 2 (圧縮する) です。
- ・ n (AllowParallelDejournaling) - 並列デジャーナリング・アップデータを実行できるミラー・メンバのタイプを決定します。指定可能な値は、0 (フェイルオーバー・メンバと災害復旧メンバ)、1 (フェイルオーバー・メンバのみ)、および 2 (すべてのメンバ) です。
- ・ o (CompressionTypeForFailoverMembers) - CompressionForFailoverMembers の圧縮タイプを決定します。指定可能な値は、0 (ZLIB)、1 (ZSTD)、および 2 (LZ4) です。
- ・ p (CompressionTypeForAsyncMembers) - CompressionForAsyncMembers の圧縮タイプを決定します。指定可能な値は、0 (ZLIB)、1 (ZSTD)、および 2 (LZ4) です。

ミラーリングの詳細は、"[ミラーリングの概要](#)" を参照してください。

管理ポータル

管理ポータルの [\[ミラーの作成\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[ミラー設定\]](#) > [\[ミラーの作成\]](#)) で、要求された情報を入力します。[\[ミラーの編集\]](#) ページで既存のミラーを編集できますが、編集操作を実行できるのはプライマリ・フェイルオーバー・メンバからのみです。

[Miscellaneous]

重要 [Miscellaneous] の各パラメータは互換性の目的でのみ保持されています。ShutdownLogErrors を除き、これらすべてのパラメータは、新しいアプリケーションの構築では使用しないでください。

このトピックでは、CPF の [Miscellaneous] セクションにある **[互換性設定]** のパラメータについて説明します。

AsyncDisconnectErr

接続切断エラーをプロセスで非同期的に受信できるようにします。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

AsyncDisconnectErr は、[DisconnectErr](#) を有効化したときの InterSystems IRIS® Data Platform の動作を変更します。AsyncDisconnectErr が有効な場合 (n = 1)、デバイスの接続切断が発生したときにプロセスは非同期で <DSCON> エラーを受信します。このエラーは、次のコマンドが実行されたときに発生し、Hang コマンドに対して割り込みが発生します。

このパラメータが有効ではない場合、次の Read コマンドまたは Write コマンドのときにプロセスが <DSCON> エラーを受信します。

AsyncDisconnectErr が適用されるのは、Windows 上の Telnet 接続のみです。それ以外のデバイス・タイプやオペレーティング・システムに対しては、効力がありません。[DisconnectErr](#) を 0 (False) に設定している場合、AsyncDisconnectErr は適用されません。

このパラメータの変更

管理ポータルでの [\[互換性設定\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[互換性\]](#)) で、[\[AsyncDisconnectErr\]](#) の行の [\[編集\]](#) を選択します。[\[AsyncDisconnectErr\]](#) を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで [CPF](#) を編集することによって、AsyncDisconnectErr を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、`%SYSTEM.Process` クラスの `AsyncDisconnectErr()` メソッドを使用します。詳細は、クラスリファレンスを参照してください。

AsyncError

非同期エラーをプロセスで受信できるようにします。

構文

```
[Miscellaneous]      AsyncError=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

AsyncError が有効な場合 (n = 1)、InterSystems IRIS® Data Platform プロセスは非同期エラーを受信できます。

このパラメータの変更

管理ポータルの **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[AsyncError]** の行の **[編集]** を選択します。**[AsyncError]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF** を編集することによって、AsyncError を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの AsyncError() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

BreakMode

プログラマ・モードでの BREAK コマンドの処理を指定します。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

InterSystems IRIS® Data Platform プログラムは、InterSystems IRIS の起動方法に応じて、2 種類のモード ([アプリケーション・モード](#)または[プログラマ・モード](#)) で実行できます。

BreakMode は、引数なしの [BREAK](#) コマンドが実行されたときに、プログラマ・モードの InterSystems IRIS プロセスがどのように処理を行うかを制御します。BreakMode が有効な場合 (n = 1)、InterSystems IRIS はデバッガを起動するか、ダイレクト・モード・プロンプトに戻って、〈BREAK〉エラーを返します。このパラメータが有効ではない場合、BREAK コマンドは無視されます。

アプリケーション・モードのジョブでは、引数のない BREAK コマンドはすべて無視されます。

このパラメータの変更

管理ポータルでの [\[互換性\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[互換性\]](#)) で、[\[BreakMode\]](#) の行の [\[編集\]](#) を選択します。[\[BreakMode\]](#) を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで [CPF](#) を[編集](#)することによって、BreakMode を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの BreakMode() メソッドを使用します。詳細は、[クラスリファレンス](#)を参照してください。

CollectResourceStats

InterSystems IRIS® データ・プラットフォームがインスタンス・リソースの統計情報を収集できるようにします。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されていますが、影響はありません。リソースの統計情報は、すべてのサポート対象プラットフォーム上ですべてのインスタンスによって必ず収集されます。

CollectResourceStats が有効な場合 (*n* = 1)、InterSystems IRIS はインスタンス・リソースの統計情報 (seize、nseize、aseize、bseize) を収集します。

インスタンス・リソースの統計情報の詳細は、“[mgstat を使用したパフォーマンスの監視](#)” と、“[Web サービスを使用した InterSystems IRIS の監視](#)” の “Enumresource” 関数を参照してください。

このパラメータの変更

管理ポータル の **[互換性]** ページ ([システム管理] > [構成] > [追加設定] > [互換性]) で、[CollectResourceStats] の行の **[編集]** を選択します。[CollectResourceStats] を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで [CPF を編集](#) することによって、CollectResourceStats を変更できます。

DisconnectErr

切断に対するプロセスの動作を設定します。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

DisconnectErr は、主入出力デバイスの切断に対する InterSystems IRIS® Data Platform の動作を決定します。このパラメータが有効な場合、ObjectScript の Write コマンドまたは Read コマンドの実行時に切断が検出されたときに、プロセスは <DSCON> エラーを受け取ります。このパラメータが有効ではない場合、切断が検知されると、プロセスはアプリケーションにエラーを通知せずに終了します。

DisconnectErr を有効にすると、主デバイスが切断された後もプロセスの実行が継続されることに注意してください。<DSCON> エラーを検出して正常に終了する処理は、アプリケーション側で実行する必要があります。DisconnectErr を有効にした場合は、以下の点に注意してください。

DisconnectErr は、TCP デバイスと、切断を検出できるターミナル・デバイスにのみ適用できます。モデムで制御されたターミナルや Windows Telnet、Windows ローカル **iristerm** (TRM:) 接続などがこの例として挙げられます。

DisconnectErr は、主デバイスにのみ適用できます。

このパラメータの変更

管理ポータルの **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、[DisconnectErr] の行の **[編集]** を選択します。[DisconnectErr] を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF を編集** することによって、DisconnectErr を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの DisconnectErr() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

FileMode

存在しないファイルへの書き込みを許可します。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

FileMode が有効な場合 (n = 1)、ファイルを開いて読み取りまたは書き込みを行おうとすると、新しいファイルが作成されます。このパラメータが有効ではない場合、新しいファイルは作成されません (ただし、[OPEN](#) コマンドのパラメータで指定されている場合を除きます)。

InterSystems IRIS® Data Platform で以下のような OPEN コマンドが実行されたとします。

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

FileMode=1 のときは、OPEN コマンドで "N" パラメータが指定されていなくても、新しいファイルが自動的に作成されます。FileMode=1 のときの結果は、OPEN コマンドに N パラメータを追加した場合と同じ結果になります。したがって、上の OPEN コマンドは以下と同じです。

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

一方、OPEN コマンドで N パラメータが指定されず、かつファイルが存在していない場合に、FileMode=0 ならば、割り込みが発生するまでプロセスが停止することを除いて、OPEN コマンドの結果は何もありません。イタリック

このパラメータの変更

管理ポータル[の](#) [\[互換性\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[互換性\]](#)) で、[\[FileMode\]](#) の行の [\[編集\]](#) を選択します。[\[FileMode\]](#) を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで [CPF](#) [を編集](#)することによって、FileMode を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの FileMode() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

GlobalKillEnabled

添え字なしのグローバルに対して KILL を実行可能にします。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

非推奨。GlobalKillEnabled が有効な場合 (n = 1)、添え字なしのグローバルに対して KILL を実行できます。したがって、グローバルの個々の添え字に対して KILL を実行しなくても、1 回の実行ですべての添え字に KILL を実行できます。このパラメータが有効ではない場合、KILL を実行すると <PROTECT> エラーが発生します。

このパラメータの変更

管理ポータル の [\[互換性\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[互換性\]](#)) で、[\[GlobalKillEnabled\]](#) の行の [\[編集\]](#) を選択します。[\[GlobalKillEnabled\]](#) を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで [CPF を編集](#) することによって、GlobalKillEnabled を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの GlobalKillDisabled() メソッドを使用します。これは GlobalKillEnabled の逆になることに注意してください。詳細は、クラスリファレンスを参照してください。

IEEEError

インスタンス全体で \$DOUBLE が INF 値および NAN 値を返すかどうかを指定します。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

IEEEError では、インスタンス全体の \$DOUBLE 関数の返り値の動作を設定します。IEEEError が有効な場合 (n = 1)、\$DOUBLE は、解決できない IEEE 浮動小数点変換に対して InterSystems IRIS® Data Platform エラーを生成します。このパラメータが有効ではない場合、\$DOUBLE は解決できない IEEE 浮動小数点の変換に対して INF (無限大)、-INF、および NAN (非数値) を返します。

このパラメータにより、\$DOUBLE 数値演算が数値に解決できない場合に INF、-INF、および NAN を発行するかどうかを制御します。どのような場合にも INF、-INF、および NAN の発行が制御されるわけではありません。\$DOUBLE は、入力値として INF、-INF、または NAN のいずれかが指定されると、このプロパティとは関係なく常にそれらの値を返します。結果が INF、-INF、または NAN になる \$DOUBLE の数値演算が、このプロパティで制御されます。これらの演算には、数値演算、指数、対数、および三角関数が含まれます。

このパラメータの変更

管理ポータルでの **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[IEEEError]** の行の **[編集]** を選択します。**[IEEEError]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF を編集** することによって、IEEEError を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの IEEEError() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

LicenseAltHeaders

クライアント・アドレスに HTTP ヘッダの代替セットを使用します。

構文

```
[Miscellaneous]    LicenseAltHeaders=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

InterSystems IRIS では、クライアント IP アドレスに既定で `remote_addr` HTTP ヘッダが使用されます。LicenseAltHeaders が真の場合、InterSystems IRIS では代わりに `HTTP_FORWARDED`、`HTTP_X_FORWARDED_FOR`、または `REMOTE_ADDR` の各ヘッダが使用されます (必要に応じて、指定の使用順序が適用されます)。

このパラメータの変更

管理ポータル の **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[LicenseAltHeaders]** の行の **[編集]** を選択します。**[LicenseAltHeaders]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、`Config.Miscellaneous` クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで [CPF を編集](#) することによって、`LicenseAltHeaders` を変更できます。

LineRecall

READ コマンドに対するコマンド行呼び出しを許可します。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

LineRecall が有効な場合 (n = 1)、READ コマンドとコマンド・プロンプトの両方で行呼び出し機能を使用できます。有効ではない場合、コマンド・プロンプトのみで行呼び出しを使用できます。

このパラメータの変更

管理ポータルの **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[LineRecall]** の行の **[編集]** を選択します。**[LineRecall]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF を編集** することによって、LineRecall を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、**%SYSTEM.Process** クラスの LineRecall() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

ListFormat

リスト内の値の圧縮形式を指定します。

構文

```
[Miscellaneous]      ListFormat=n
```

n は、0 から 3 の範囲の整数です。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

ListFormat では、リスト内で圧縮する必要がある値を指定します。ListFormat に指定できるオプションは以下のとおりです。

- ・ 0 – リスト内で圧縮しません。
- ・ 1 – リスト内の [\\$DOUBLE](#) (IEEE) 値を圧縮します。
- ・ 2 – リスト内の Unicode 文字列を圧縮します。
- ・ 3 – リスト内の [\\$DOUBLE](#) と Unicode 文字列の両方を圧縮します。

注釈 外部クライアント (Java、C# など) でリストを使用する場合は、その外部クライアントが圧縮されたリスト形式をサポートしていることを確認してください。

このパラメータの変更

管理ポータルの [\[互換性\]](#) ページ ([\[システム管理\]](#)→[\[構成\]](#)→[\[追加の設定\]](#)→[\[互換性\]](#)) で、[\[ListFormat\]](#) の行の [\[編集\]](#) を選択します。この設定の目的の値を入力します。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで [CPF](#) を[編集](#)することによって、ListFormat を変更できます。

関連項目

- ・ [\\$DOUBLE](#) 関数
- ・ [\\$LIST](#) 関数

LogRollback

トランザクション・ロールバックをログに記録することを許可します。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

LogRollback が有効な場合 (n = 1)、InterSystems IRIS® Data Platform は、トランザクション・ロールバックを **messages.log** ファイルに記録します (このファイルは、install-dir¥mgr ディレクトリ、または **console** パラメータで指定された代替ディレクトリにあります)。LogRollback が有効ではない場合、トランザクション・ロールバックは記録されません。

このパラメータの変更

管理ポータル の **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[LogRollback]** の行の **[編集]** を選択します。**[LogRollback]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF** を編集することによって、LogRollback を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの LogRollback() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

MVDefined

未使用。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

未使用。

NodeNameInPid

InterSystems IRIS® Data Platform が特殊変数 \$JOB を参照するときの振る舞いを指定します。

構文

```
[Miscellaneous]      NodeNameInPid=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

NodeNameInPid が有効な場合 (n = 1)、\$JOB は、現在のプロセスのプロセス ID 番号とノード名を連結したものを返します。NodeNameInPid が有効ではない場合、\$JOB は、ノード名の付かないプロセス ID 番号のみを返します。

このパラメータの変更

管理ポータルの **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[NodeNameInPid]** の行の **[編集]** を選択します。**[NodeNameInPid]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF を編集** することによって、NodeNameInPid を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、**%SYSTEM.Process** クラスの NodeNameInPid() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

NullSubscripts

グローバル参照での NULL 添え字の使用を許可します。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

NullSubscripts が有効な場合 (n = 1)、グローバル参照に対して NULL 添え字を使用できます。このパラメータが有効ではない場合、NULL 添え字を使用すると <SUBSCRIPT> エラーが発生します。このパラメータは無効のままにすることをお勧めします。

このパラメータの変更

管理ポータルでの **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[NullSubscripts]** の行の **[編集]** を選択します。**[NullSubscripts]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF を編集** することによって、NullSubscripts を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの NullSubscripts() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

OldZU5

ネームスペースの切り替え時に、グローバル・ベクトルを消去するかどうかを指定します。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

OldZU5 が有効な場合 (n = 1)、[ZN](#) コマンドを使用して現在のネームスペースに切り替えると、グローバル・ベクトルのキャッシュが消去されます。このパラメータが有効ではない場合、現在のネームスペースに切り替えても効果はありません。

このパラメータの変更

管理ポータル の [\[互換性\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[互換性\]](#)) で、[\[OldZU5\]](#) の行の [\[編集\]](#) を選択します。[\[OldZU5\]](#) を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで [CPF](#) を[編集](#)することによって、OldZU5 を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、`%SYSTEM.Process` クラスの `OldZU5()` メソッドを使用します。詳細は、クラスリファレンスを参照してください。

OpenMode

シーケンシャル・ファイルを開くときに使用する読み取り/書き込みモードを指定します。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

OpenMode は、[OPEN](#) コマンドでシーケンシャル・ファイルを開くときに使用する既定の読み取り/書き込みモードを指定します。オプションは [\[読み込み/書き込み\]](#) (1) または [\[読み込み\]](#) (0) です。

このパラメータの変更

管理ポータルの [\[互換性\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[互換性\]](#)) で、[\[OpenMode\]](#) の行の [\[編集\]](#) を選択します。[\[読み込み\]](#) (0) または [\[読み込み/書き込み\]](#) (1) のいずれかのモードを選択します。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで [CPF を編集](#) することによって、OpenMode を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの OpenMode() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

PopError

エラー・ハンドラをスタックからポップ・オフするタイミングを指定します。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

PopError が有効な場合 (n = 1)、エラーがトリガされたときに、InterSystems IRIS® Data Platform によって現在のエラー・ハンドラがスタックからポップ・オフされます。この場合、インスタンスによって \$ZTRAP エラー・ハンドラが呼び出されると、エラー・ハンドラはスタックから削除されます。そのため、エラー・ハンドラの実行中にエラーが発生した場合は、そのエラーはスタックの前のエラー・ハンドラによって処理されます。

このパラメータが有効ではない場合、通常の振る舞いが優先します。エラー・ハンドラが呼び出されても \$ZTRAP エラー・ハンドラはアクティブなままです。この場合、インスタンスによって \$ZTRAP エラー・ハンドラが呼び出されると、エラー・ハンドラは作成されたエラー・ハンドラのスタックに残ります。このため、エラー・ハンドラの実行中にエラーが発生すると、そのエラー・ハンドラが自分を呼び出そうとして同じエラーが再度返され、エラー・ハンドラが \$ZTRAP を明示的に新しい値に設定するまで無限ループとなります。

このパラメータの変更

管理ポータル の **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[PopError]** の行の **[編集]** を選択します。**[PopError]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF を編集** することによって、PopError を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの PopError() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

RefInKind

\$NAME および \$QUERY が拡張グローバル参照を処理する方法を指定します。

構文

```
[Miscellaneous]      RefInKind=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

RefInKind が有効な場合 (n = 1)、[拡張グローバル参照](#)である入力を指定すると、[\\$NAME](#) および [\\$QUERY](#) は、拡張参照なしでグローバル名のみを返します。このパラメータが有効ではない場合、これらの関数は拡張グローバル参照を返します。

このパラメータの変更

管理ポータルの [\[互換性\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[互換性\]](#)) で、[\[RefInKind\]](#) の行の [\[編集\]](#) を選択します。[\[RefInKind\]](#) を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで [CPF を編集](#)することによって、RefInKind を変更できます。

注釈 このパラメータの値を変更した場合、変更内容は、変更時に実行されていたプロセスではなく、変更後に起動したプロセスに適用されます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、`%SYSTEM.Process` クラスの `RefInKind()` メソッドを使用します。詳細は、[クラスリファレンス](#)を参照してください。

ScientificNotation

インスタンス全体で科学的記数法の記号として小文字“e”を使用できるようにします。

構文

```
[Miscellaneous]      ScientificNotation=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

ScientificNotation が有効な場合 (n = 1)、InterSystems IRIS® データ・プラットフォームは、インスタンス全体で科学的記数法の記号として小文字“e”を使用します。

このパラメータの変更

管理ポータルでの **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[ScientificNotation]** の行の **[編集]** を選択します。**[ScientificNotation]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF を編集** することによって、ScientificNotation を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、**%SYSTEM.Process** クラスの ScientificNotation() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

SetZEOF

シーケンシャル・ファイルの読み取り時に予期しないファイルの終了エラーが発生した場合の振る舞いを指定します。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

SetZEOF が有効な場合 (*n* = 1)、InterSystems IRIS® Data Platform によって、シーケンシャル・ファイルの最後に到達したことを示す特殊変数 \$ZEOF が設定されます。このパラメータが有効ではない場合、代わりに <ENDOFFILE> エラーがスローされます。

このパラメータの変更

管理ポータル の **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[SetZEOF]** の行の **[編集]** を選択します。**[SetZEOF]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF を編集** することによって、SetZEOF を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの SetZEOF() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

ShutDownLogErrors

シャットダウン時に InterSystems IRIS® Data Platform のシステム・エラー・ログ・エントリを messages.log ファイルに書き込むことを許可します。

構文

```
[Miscellaneous]      ShutDownLogErrors=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

ShutDownLogErrors が有効な場合 (n = 1)、InterSystems IRIS は、^SYSLOG からのエラー情報を **messages.log** ファイルに記録します (このファイルは、install-dir¥mgr ディレクトリ、または **console** パラメータで指定された代替ディレクトリにあります)。ShutDownLogErrors が有効ではない場合、これらのエラーは記録されません。

詳細は、“[InterSystems IRIS システム・エラー・ログ](#)” を参照してください。

このパラメータの変更

管理ポータル の **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[ShutDownLogErrors]** の行の **[編集]** を選択します。**[ShutDownLogErrors]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF** を編集することによって、ShutDownLogErrors を変更できます。

StopID

未使用。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

未使用。

SwitchOSDir

ネームスペースの変更時に、現在の作業ディレクトリを変更することを禁止します。

構文

```
[Miscellaneous]      SwitchOSDir=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

SwitchOSDir は、別のネームスペースに切り替えた場合に (相対パス名でファイルにアクセスする場合など)、現在の作業ディレクトリをどのように処理するかを指定します。SwitchOSDir が有効な場合 (n = 1)、ネームスペースを変更するときに、切り替えるネームスペースに関係なく、現在の作業ディレクトリは変わりません。

このパラメータが有効ではない場合、ネームスペースを変更すると、現在の作業ディレクトリは、新しいネームスペースで % のないグローバルの既定のデータセットのディレクトリに変わります。ただし、このデータセットがリモートの場合 (別のシステムにネットワーク接続されている場合)、現在の作業ディレクトリは変わりません。

例えば、SwitchOSDir が 1 に設定されている場合、または SwitchOSDir が 0 に設定されていてデータセットがリモート場合があります。その場合、ネームスペースを変更した後、現在の作業ディレクトリは自動的に変わりませんが、プログラムでいつでも現在の作業ディレクトリを変更できます。

このパラメータの変更

管理ポータル の **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[SwitchOSDir]** の行の **[編集]** を選択します。**[SwitchOSDir]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF** を編集することによって、SwitchOSDir を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの SwitchOSDir() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

SynchCommit

TCOMMIT と対応するジャーナル書き込み操作の同期を無効にします。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

TCOMMIT コマンドを実行するたびに、トランザクションに伴うジャーナル・データをディスクへフラッシュするよう要求されます。**SynchCommit** は、この要求時の振る舞いを制御するものです。有効な場合 (n = 1)、TCOMMIT はジャーナル・データの書き込み操作が完了した後に完了します。**SynchCommit** が有効ではない場合、TCOMMIT は書き込み操作を待たずに完了します。

このパラメータの変更

管理ポータル の **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[SynchCommit]** の行の **[編集]** を選択します。**[SynchCommit]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、**Config.Miscellaneous** クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF** を編集することによって、**SynchCommit** を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、**%SYSTEM.Process** クラスの **SynchCommit()** メソッドを使用します。詳細は、クラスリファレンスを参照してください。

TelnetNUL

Telnet 転送で行末の Telnet NUL を抑制します。Windows システムでのみ有効です。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

出力の際、Telnet ネットワーク仮想ターミナル (NVT) は既定の行末処理を実行します。具体的には、キャリッジ・リターン文字 (CR) に続いて改行文字 (LF) を発行するか、CR に続いて NUL (LF が発行されない場合) を発行します。TelnetNUL は、後者のケースで NUL 文字を発行するかどうかを制御します。TelnetNul が有効な場合 (*n* = 1)、Telnet 仮想ターミナルは NUL 文字を抑制します。

この設定は Windows システムのみに適用され、Telnet がオペレーティング・システム・ベンダから提供されている UNIX®、および Linux の構成では無視されます。

このパラメータの変更

管理ポータルでの **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[TelnetNUL]** の行の **[編集]** を選択します。**[TelnetNUL]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF** を編集することによって、TelnetNUL を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの TelnetNUL() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

TruncateOverflow

数値オーバーフローのときに <MAXNUMBER> エラーを抑止します。

構文

```
[Miscellaneous]      TruncateOverflow=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

通常、InterSystems IRIS® Data Platform では、極端に大きな数字 (約 1.0E147 または -1.0E146) が見つかると、<MAXNUMBER> エラーがスローされます。TruncateOverflow が有効な場合、<MAXNUMBER> エラーが抑制されます。

詳細は、“[極端に大きな数字](#)” を参照してください。

このパラメータの変更

管理ポータルの **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[TruncateOverflow]** の行の **[編集]** を選択します。**[TruncateOverflow]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF を編集** することによって、TruncateOverflow を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの TruncateOverflow() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

関連項目

- ・ [\\$DOUBLE](#) 関数

Undefined

ObjectScript が、存在しない変数をフェッチしようとしたときの動作を指定します。

構文

```
[Miscellaneous]      Undefined=n
```

n は、0、1、または 2 です。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

パラメータ Undefined は、ObjectScript が、未定義の変数の値をフェッチしようとしたときの動作を指定します。Undefined の値は 0、1、または 2 です。

- ・ 0 - 常に <UNDEFINED> エラーをスローします (既定)。
- ・ 1 - 未定義の変数に添え字がある場合は NULL 文字列を返しますが、未定義の変数が単一値の場合は <UNDEFINED> エラーをスローします。
- ・ 2 - 常に NULL 文字列を返します。

このパラメータの変更

管理ポータル の **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、[Undefined] の行の **[編集]** を選択します。目的のオプションを選択します。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF を編集** することによって、Undefined を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの Undefined() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

UseNagleAlgorithm

InterSystems IRIS® Data Platform で Telnet に Nagle アルゴリズムを使用できるようにします。

構文

```
[Miscellaneous]      UseNagleAlgorithm=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

UseNagleAlgorithm が有効な場合 (n = 1)、InterSystems IRIS は Telnet に Nagle アルゴリズムを使用します。

Nagle アルゴリズムを利用すると、Telnet の効率が向上します。このアルゴリズムは、一定の短い時間内に送信される複数のメッセージを 1 つの IP パケットにまとめることで、ネットワークで送信される IP パケットの数を減らします。Nagle アルゴリズムが有効化されているときは、送信コマンドからのデータを実際にコミットする前に、オペレーティング・システムはしばらくの間待機します。これは、その間にアプリケーションから再びデータ送信が呼び出された場合、そのデータを最初のデータと共にまとめることができるためです。詳細は、“[RFC 896](#)” を参照してください。

このパラメータの変更

管理ポータルでの **[互換性]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[互換性]**) で、**[UseNagleAlgorithm]** の行の **[編集]** を選択します。**[UseNagleAlgorithm]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで **CPF** を編集することによって、UseNagleAlgorithm を変更できます。

ViewPastData

\$VIEW で InterSystems IRIS® Data Platform メモリ領域外のデータを検証できるようにします。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

ViewPastData が有効な場合 (n = 1)、[\\$VIEW](#) コマンドを使用して、InterSystems IRIS メモリ領域外のデータを検証できます。このパラメータが有効ではない場合、\$VIEW コマンドはエラーをスローします。

このパラメータの変更

管理ポータル[の](#) [\[互換性\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[互換性\]](#)) で、[\[ViewPastData\]](#) の行の [\[編集\]](#) を選択します。[\[ViewPastData\]](#) を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで [CPF](#) を[編集](#)することによって、ViewPastData を変更できます。

ZDateNull

無効な値に対する \$ZDATE の応答を指定します。

構文

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

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

ZDateNull は、無効な値によって [\\$ZDATE](#) が呼び出された場合の応答を決定します。このパラメータが有効な場合、\$ZDATE は NULL 値を返します。このパラメータが有効ではない場合は、エラーを返します。

このパラメータの変更

管理ポータルの [\[互換性\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[互換性\]](#)) で、[\[ZDateNull\]](#) の行の [\[編集\]](#) を選択します。[\[ZDateNull\]](#) を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Miscellaneous クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで [CPF を編集](#) することによって、ZDateNull を変更できます。

(インスタンス全体ではなく) 1 つのプロセスに対してのみこのパラメータを変更するには、%SYSTEM.Process クラスの ZDateNull() メソッドを使用します。詳細は、クラスリファレンスを参照してください。

[Monitor]

このトピックでは、CPF の [Monitor] セクションのパラメータについて説明します。

SNMPEnabled

Simple Network Management Protocol (SNMP) が自動的に開始できるようにします。

構文

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

n は 1 または 0 です。既定値は 0 です。

説明

SNMPEnabled が有効な場合 (*n* = 1)、InterSystems IRIS® Data Platform が起動したときに自動的に SNMP エージェントが起動します。

SNMP 監視を有効にするには、**[システム開始時にSNMPエージェントを開始]** を選択します。また、**[サービス]** ページ (**[システム管理]** > **[セキュリティ]** > **[サービス]**) で **%Service_Monitor** を有効にしておく必要もあります。

このパラメータの変更

管理ポータル **[モニタ]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[モニタ]**) で、**[システム開始時にSNMPエージェントを開始]** を選択して、この設定を有効にします。**%Service_Monitor** を有効にするためのショートカットもあります。

管理ポータルを使用する代わりに、Config.Monitor クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (**“アクティブな CPF の編集”** を参照)、SNMPEnabled を変更できます。

関連項目

- “監視ガイド” の “[SNMP を使用した InterSystems IRIS の監視](#)”

[Namespaces]

このトピックでは、CPF の [Namespaces] セクションのパラメータについて説明します。

Namespace

InterSystems IRIS® Data Platform ネームスペースを定義します。

構文

```
[Namespaces]      Name=globals,routines,temporary
```

説明

[Namespaces] セクションには、InterSystems IRIS インスタンスに対して定義されているすべてのネームスペースのエントリが含まれます。管理ポータルを使用してネームスペースの追加および構成を行うと、構成パラメータ・ファイルにエントリが自動的に追加されます。

各エントリにはコンマ区切り値が最大で3つ含まれますが、必須の値は最初の値のみです。その他の値は、値を指定しない場合、インスタンスの既定値に設定されます。この値は以下のとおりです。

- globals – グローバル (一時グローバル以外) の既定のデータベース名。必須項目。
- routines – ルーチンおよびクラスに対する既定のデータベース名。データベースを指定しない場合は、既定でグローバル・データベースに設定されます。
- temporary – 一時的なストレージ (特に一時グローバルのストレージ) に対する既定のデータベース名。データベースが指定されていない場合、既定値は **IRISTEMP** です。

一時グローバルは、名前が ^IRIS.Temp (大文字小文字を区別する) で始まるグローバルです。

既定値の例外

先頭が % である名前を持つグローバルは、ユーザ定義グローバル・マッピングによって別のデータベースにマッピングされている場合を除き、**IRISSYS** にマッピングされます。

すべてのネームスペースで、% で始まるルーチンおよびクラスは、**IRISLIB** データベースから得られます。ただし、次の例外があります。

- 先頭が %SYS.* である名前を持つルーチンおよびクラスは、**IRISSYS** から得られます (InterSystems IRIS に付属するルーチンおよびクラス)。
- 先頭が %Z* または %z* である名前を持つルーチンおよびクラスは、**IRISSYS** から得られます (ユーザ定義のルーチンおよびクラス)。
- ルーチンまたはパッケージのマッピングを使用して、ユーザが別のデータベースから明示的にマッピングするルーチンおよびクラス。

IRISSYS データベースに配置されているルーチンには、特別なセキュリティ特権があることに注意してください。例えば、そのルーチンを実行するプロセスのロールやその他のセキュリティ属性を変更できます。

例

[Namespaces] セクションでは、以下のように、エントリごとにその全体が 1 行で記述されます。

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

このパラメータの変更

管理ポータルの [ネームスペース] ページ ([システム管理] > [構成] > [システム構成] > [ネームスペース]) に移動します。新しいエントリを追加するには、[新規ネームスペース作成] を選択します。既存のエントリを編集するには、そのエントリの行の [編集] を選択します。

関連項目

このドキュメントの [Map] セクションの [Global](#) エントリ

このドキュメントの [Map] セクションの [Package](#) エントリ

このドキュメントの [Map] セクションの [Routine](#) エントリ

[SQL]

このトピックでは、管理ポータルの [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) の設定について説明します。このページの下部には、CPF の [SQL] セクションにあるパラメータのリストが記載されています。

[SQL] ページは、[SQL タブ] と [SQL シェル] タブに分かれています。[SQL タブ] では、次の表に示す CPF パラメータに対応するさまざまな SQL 設定を構成できます。[SQL シェル] タブのオプションについては、“SQL シェル・インタフェースの使用法” の “SQL シェルの構成” で説明しています。

テーブル W-1: SQL タブ

SQL タブの設定	CPF の同等のパラメータ
クエリキャッシュのソースを保持	SaveMAC
GETDATE()、CURRENT_TIME、および CURRENT_TIMESTAMP の既定の時間精度	TimePrecision
ロックエスカレーション閾値	LockThreshold
TO_DATE の既定の形式	TODATEDefaultFormat
VARCHAR のデフォルト長	ODBCVarcharMaxlen
デフォルトスキーマ	DefaultSchema
単一プロセス内でクエリを実行	AutoParallel
DDL を使用して作成されたテーブルのプライマリーキーをIDキーとして定義する	IdKey
冗長な DDL ステートメントを無視	DDLNo* パラメータを設定
GROUP BY と DISTINCT クエリでオリジナルの値を結果に返す	FastDistinct
アダプティブモードをオフにして実行時プラン選択と自動チューニングを無効にする	AdaptiveMode
クエリ実行のパラメータを収集するパラメータ収集を有効にする	ParameterSampling
ロック・タイムアウト (秒)	LockTimeout
クライアント接続の TCP キープアライブ(秒)	TCPKeepAlive
Client maximum idle time (seconds)	ClientMaxIdleTime

ANSIPrecedence

SQL クエリの演算子の優先順位を指定します。

構文

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

n は 1 または 0 です。既定値は 1 です。

説明

ANSIPrecedence が有効な場合 (n = 1)、InterSystems SQL は算術演算子に関する ANSI の優先順位を使用します。ANSIPrecedence が無効な場合 (n = 0)、InterSystems SQL は算術式を必ず左から右の順番で実行します。これは、インスタンス全体の構成設定です。

ANSI の優先順位を構成した場合、演算子 “*”、“¥”、“/”、および “#” は、演算子 “+”、“-”、および “||” よりも優先順位が高くなります。優先順位の高い演算子は、優先順位の低い演算子より先に実行されます。この優先順位は、必要に応じて括弧を使用することでオーバーライドできます。

詳細は、“InterSystems SQL の使用法” の “[演算子の優先順位](#)” のセクションを参照してください。

このパラメータの変更

ターミナルから ANSIPrecedence に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“ANSIPrecedence”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、ANSIPrecedence を変更できます。

関連項目

- “InterSystems SQL の使用法” の “言語要素” の章にある “[算術演算子と算術関数](#)” のセクション

AdaptiveMode

アダプティブ・モードのパフォーマンス最適化オプションを有効にします。

構文

```
[SQL]      AdaptiveMode=n
```

n は 1 または 0 です。既定値は 1 です。

説明

AdaptiveMode が有効 (n = 1) である場合、InterSystems SQL ではアダプティブ・モードが使用されます。これにより、可能な範囲で自動化とアダプティブの設定が有効になります。アダプティブ・モードでは以下のオプションを構成します。

- ・ InterSystems IRIS® データ・プラットフォームを新しいメジャー・バージョンにアップグレードする際に、既存のクエリ・プランの凍結を無効にします。アップグレード時に、凍結されていないクエリ・プランは無効になり、アップグレード後に初めて呼び出されるときに再計画されます。詳細は、“[凍結プラン](#)”を参照してください。
- ・ インスタンス全体で並列処理を可能にします。詳細は、“[並列クエリ処理の構成](#)”を参照してください。
 - － [ParallelProcessing](#) オプションが無効 (0 に設定) の場合、アダプティブ・モードによってこの設定がオーバーライドされ、並列処理が有効になります。
- ・ チューニングしていないテーブルを最初にクエリすると、自動テーブル・チューニングが有効になります。テーブルをチューニングすると、今後のクエリの最適化に使用できるテーブルから統計情報が収集されます。詳細は、“[テーブルのチューニング](#)”を参照してください。
- ・ 実行時プラン選択を有効にします。これにより、InterSystems SQL では実行時のクエリ・パラメータの値を考慮して (異常値の存在も含め) 代替案が検討されます。その結果、このパラメータを有効にすると、以前の [BiasQueriesAsOutlier](#) と [RTPC](#) の設定がオーバーライドされます。

このパラメータの変更

管理ポータル の [SQL] ページ ([システム管理] > [構成] > [SQL とオブジェクトの設定] > [SQL]) で [アダプティブモードをオフにして実行時プラン選択と自動チューニングを無効にする] を選択してアダプティブ・モードを無効にします。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アダプティブな CPF の編集](#)”を参照)、AdaptiveMode を変更できます。

AllowRowIDUpdate

ユーザが RowID 値を更新できるようにします。

構文

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

n は 1 または 0 です。既定値は 0 です。

説明

AllowRowIDUpdate が有効な場合 (n = 1)、RowID 値はユーザが変更可能です。RowID 値の変更は、取り返しの付かない結果を招くことがあるため、極めて特別な状況でのみ、十分な注意を払って行う必要があります。1 に設定するのは、BEFORE トリガで独自のファイリングを行い、%SkipFiling フラグを使用する場合のみです。それ以外の場合は、既定値の 0 を使用してください。

このパラメータの変更

Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、AllowRowIDUpdate を変更できます。

関連項目

- ・ “テーブルの定義” の章の "[RowID フィールド](#)” セクション。

AutoParallel

インスタンス全体で並列処理を許可します。

構文

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

n は 1 または 0 です。既定値は 1 です。

説明

AutoParallel が有効な場合 (n = 1)、InterSystems SQL クエリで[並列処理](#)を使用して実行効率を向上させられます。これは、シャード環境ではすべてのクエリが並列処理を使用して実行されることを意味します。非シャード環境では、並列処理を使用するかどうかは、[AutoParallelThreshold](#) の値に基づいてクエリごとに判断されます。

このパラメータが有効ではない場合、すべてのクエリは単一プロセスで実行されます。

[AdaptiveMode](#) が有効 (1 に設定) で AutoParallel が無効な場合、アダプティブ・モードによって AutoParallel の設定がオーバーライドされ、並列処理が有効になります。

このパラメータの変更

管理ポータル の [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[単一プロセスでクエリを実行する] を選択して、並列処理を無効にします。

ターミナルから AutoParallel に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“AutoParallel”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、AutoParallel を変更できます。

AutoParallelThreshold

並列処理のしきい値を設定します。

構文

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

n は負ではない任意の整数です。既定値は 3200 です。

説明

n の値が高いほど、InterSystems SQL クエリが[並列処理](#)を使用して実行される可能性は低くなります。値 n は、アクセスしたマップで並列処理を実行するために必要なタプルの最小数にほぼ相当します。

[AutoParallel](#) が無効な場合、AutoParallelThreshold には何の効果也没有せん。

このパラメータの変更

ターミナルから AutoParallelThreshold に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“AutoParallelThreshold”) メソッドを使用します。詳細は、[クラスリファレンス](#)を参照してください。

あるいは、Config.SQL クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)”を参照)、AutoParallelThreshold を変更できます。

BiasQueriesAsOutlier

クエリ最適化を異常値優先に設定します。

構文

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

n は 1 または 0 です。既定値は 0 です。

説明

BiasQueriesAsOutlier が有効な場合 (n = 1)、InterSystems SQL は、主に異常値を返すクエリに対して最適化されます。異常値の選択性の詳細は、“InterSystems SQL 最適化ガイド”の“テーブルの最適化”の章にある“[テーブルのチューニング](#)”を参照してください。

BiasQueriesAsOutlier と RTPC の両方を同時に有効にすることはできません。BiasQueriesAsOutlier と RTPC の両方を 1 に設定すると RTPC が有効になり、BiasQueriesAsOutlier 設定は無視されます。RTPC が設定されている場合、InterSystems SQL は、クエリごとに異常値の最適化を使用できます。

[AdaptiveMode](#) が有効 (1 に設定) で、BiasQueriesAsOutlier がアクティブで有効な場合、アダプティブ・モードによって BiasQueriesAsOutlier の設定がオーバーライドされ、異常値バイアスが無効になります。

注釈 アダプティブ・モードは以前の BiasQueriesAsOutlier メカニズムに代わる機能です。この設定を以前に使用していて、サポートが必要な場合は、インターシステムズのサポート窓口までご連絡ください。

このパラメータの変更

Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)”を参照)、BiasQueriesAsOutlier を変更できます。

ClientMaxIdleTime

構文

```
[SQL]      ClientMaxIdleTime=n
```

n は、クライアントとサーバ間のアイドル状態の接続を強制的に切断するまでの許容秒数です。既定値は 0 です。

説明

ClientMaxIdleTime が 0 以外の場合、InterSystems SQL は指定された秒数間待機してから、クライアントとサーバ間のアイドル状態の接続を閉じます。このタイムアウトは、JDBC、ODBC、ADO.NET を含め、アイドル状態になっているすべての SQL クライアントに適用されます。

このパラメータの変更

管理ポータルでの [SQL] ページ ([システム管理] > [構成] > [SQL とオブジェクトの設定] > [SQL]) で、[Client maximum idle time (seconds)] 設定に値を入力します。

SetOption() オプションを使用して、またはテキスト・エディタで CPF を編集して ("[アクティブな CPF の編集](#)" を参照)、ClientMaxIdleTime を変更できます。また、管理ポータルでの [SQL] ページ ([システム管理] > [構成] > [SQL とオブジェクトの設定] > [SQL]) で、[Client maximum idle time (seconds)] 設定に値を入力してこのパラメータを変更できます。

Comment

埋め込み SQL 文を、コメントとしてソース・コード内に保持します。

構文

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

n は 1 または 0 です。既定値は 1 です。

説明

Comment が有効な場合 (n = 1)、埋め込み SQL 文は、ルーチンのソース・コード (.INT) のコメントとして保持されます。

このパラメータの変更

ターミナルから Comment に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“RetainSQL”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、Comment を変更できます。

DBMSSecurity

SQL セキュリティを有効にします。

構文

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

n は 1 または 0 です。既定値は 1 です。

説明

DBMSSecurity が有効な場合、すべての InterSystems SQL セキュリティが有効になります。これは、特権ベースのテーブルやビュー、プロシージャのセキュリティがアクティブであることを意味します。ユーザは、特権が付与されているテーブルやビューのみで、表示やアクションを実行できます。このパラメータが有効ではない場合、ユーザは、必要な特権がなくても、テーブルに対して表示やアクションを実行できます。

詳細は、“[GRANT](#)”、“[CREATE TABLE](#)”、および “[CREATE VIEW](#)” コマンドを参照してください。

このパラメータの変更

ターミナルから DBMSSecurity に目的の値を設定するには、`%SYSTEM.SQL.Util` クラスの `SetOption(“SQLSecurity”)` メソッドを使用します。詳細は、[クラスリファレンス](#)を参照してください。このオプションの変更は、新規プロセスのみに適用され、既存のプロセスには適用されません。

あるいは、`Config.SQL` クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)”を参照)、DBMSSecurity を変更できます。

DDLDefineBitmapExtent

DDL 文によって作成されたテーブルでビットマップ・エクステント・インデックスを定義するかどうかを指定します。

構文

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

n は 1 または 0 です。既定値は 1 です。

説明

DDLDefineBitmapExtent が有効な場合 (n = 1)、DDL CREATE TABLE 文によって作成されたテーブルでビットマップ・エクステント・インデックスが定義されます。このインデックスにより、[COUNT\(*\)](#) (テーブル内の行数を返す関数) のパフォーマンスが向上します。

このパラメータの変更

ターミナルから DDLDefineBitmapExtent に目的の値を設定するには、`%SYSTEM.SQL.Util` クラスの `SetOption` (“DDLDefineBitmapExtent”) メソッドを使用します。詳細は、[クラスリファレンス](#)を参照してください。

あるいは、`Config.SQL` クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)”を参照)、DDLDefineBitmapExtent を変更できます。

DDLFinal

DDL 文によって作成されたクラスを Final にするかどうかを指定します。

構文

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

n は 1 または 0 です。既定値は 1 です。

説明

DDLFinal が有効な場合 (n = 1)、DDL CREATE TABLE 文によって作成されたテーブルは [Final](#) になります。つまり、サブクラスを持つことはできません。

このパラメータの変更

ターミナルから DDLFinal に目的の値を設定するには、`%SYSTEM.SQL.Util` クラスの `SetOption("DDLFinal")` メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、`Config.SQL` クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも ("[アクティブな CPF の編集](#)" を参照)、DDLFinal を変更できます。

DDLNo201

既存のテーブルの CREATE を実行して発生するエラーを抑制します。

構文

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

n は 1 または 0 です。既定値は 0 です。

説明

DDLNo201 が有効な場合 (n = 1)、既存のテーブルまたはビューの CREATE を実行しようとしたとき、SQLCODE -201 エラーが InterSystems IRIS® Data Platform によって抑制されます。このパラメータが有効ではない場合、InterSystems IRIS はエラーを返します。

詳細は、“[CREATE TABLE](#)” コマンドおよび “[CREATE VIEW](#)” コマンドを参照してください。

このパラメータの変更

管理ポータル の [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[冗長な DDL 文を無視] を選択して、冗長な SQLCODE エラーを抑制します。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、DDLNo201 を変更できます。

DDLNo30

存在しないテーブルの DROP を実行して発生するエラーを抑制します。

構文

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

n は 1 または 0 です。既定値は 0 です。

説明

DDLNo30 が有効な場合 (n = 1)、存在しないテーブルの DROP を実行しようとしたとき、SQLCODE -30 エラーが InterSystems IRIS® Data Platform によって抑制されます。このパラメータが有効ではない場合、InterSystems IRIS はエラーを返します。

詳細は、“[DROP TABLE](#)” コマンドおよび “[DROP VIEW](#)” コマンドを参照してください。

このパラメータの変更

管理ポータル の [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[冗長な DDL 文を無視] を選択して、冗長な SQLCODE エラーを抑制します。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、DDLNo30 を変更できます。

DDLNo307

既存の主キー制約の CREATE を実行して発生するエラーを抑制します。

構文

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

n は 1 または 0 です。既定値は 0 です。

説明

DDLNo307 が有効な場合 (*n* = 1)、既に主キー制約が存在するテーブルに対して、DDL の CREATE を使って主キー制約を作成しようとしたとき、SQLCODE -307 エラーが InterSystems IRIS® Data Platform によって抑制されます。

詳細は、“[CREATE TABLE](#)” コマンドおよび “[ALTER TABLE](#)” コマンドを参照してください。

このパラメータの変更

管理ポータルの [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[冗長な DDL 文を無視] を選択して、冗長な SQLCODE エラーを抑制します。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、DDLNo307 を変更できます。

DDLNo311

ADD による外部キーの追加時に、その名前のキーが既に存在していたとき、エラーを抑制します。

構文

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

n は 1 または 0 です。既定値は 0 です。

説明

DDLNo311 が有効な場合 (n = 1)、ADD で外部キーを追加しようとしたときに、その名前のキーが既に存在していても、SQLCODE -311 エラーは InterSystems IRIS® Data Platform によって抑制されます。

詳細は、“[ALTER TABLE](#)” コマンドを参照してください。

このパラメータの変更

管理ポータルの [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[冗長な DDL 文を無視] を選択して、冗長な SQLCODE エラーを抑制します。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、DDLNo311 を変更できます。

DDLNo315

存在しない制約の DROP を実行して発生するエラーを抑制します。

構文

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

n は 1 または 0 です。既定値は 0 です。

説明

DDLNo315 が有効な場合 (*n* = 1)、存在しない制約の DROP を実行しようとする、SQLCODE -315 エラーが InterSystems IRIS® Data Platform によって抑制されます。

詳細は、“[ALTER TABLE](#)” コマンドを参照してください。

このパラメータの変更

管理ポータル の [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[冗長な DDL 文を無視] を選択して、冗長な SQLCODE エラーを抑制します。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、DDLNo315 を変更できます。

DDLNo324

既存のインデックスの CREATE を実行して発生するエラーを抑制します。

構文

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

n は 1 または 0 です。既定値は 0 です。

説明

DDLNo324 が有効な場合 (n = 1)、既存のインデックスの CREATE を実行しようとする、SQLCODE -324 エラーが InterSystems IRIS® Data Platform によって抑制されます。

詳細は、“[CREATE INDEX](#)” コマンドを参照してください。

このパラメータの変更

管理ポータルの [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[冗長な DDL 文を無視] を選択して、冗長な SQLCODE エラーを抑制します。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、DDLNo324 を変更できます。

DDLNo333

存在しないインデックスの DROP を実行して発生するエラーを抑制します。

構文

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

n は 1 または 0 です。既定値は 0 です。

説明

DDLNo333 が有効な場合 (n = 1)、存在しないインデックスの DROP を実行しようとする、SQLCODE -333 エラーが InterSystems IRIS® Data Platform によって抑制されます。

詳細は、“[DROP INDEX](#)” コマンドを参照してください。

このパラメータの変更

管理ポータル の [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[冗長な DDL 文を無視] を選択して、冗長な SQLCODE エラーを抑制します。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、DDLNo333 を変更できます。

DDLSQLOnlyCompile

SQL のみのコンパイルを可能にします。

構文

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

n は 1 または 0 です。既定値は 0 です。

説明

DDLSQLOnlyCompile が有効な場合 (*n* = 1)、DDL 文を実行した結果としてクラスを対象に実行されるコンパイルで、*q* (sqlonly) フラグを設定したクラスがコンパイルされます。このパラメータが有効ではない場合、*q* フラグは使用されません。

このパラメータの変更

Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、DDLSQLOnlyCompile を変更できます。

DDLUseExtentSet

DDL 文によって作成されたテーブルのインデックス・データを格納するグローバルに、ハッシュ化した名前を使用できるようにします。

構文

```
[SQL]      DDLUseExtentSet=n
```

n は 1 または 0 です。既定値は 1 です。

説明

DDLUseExtentSet が有効な場合 (*n* = 1)、DDL CREATE TABLE 文によって作成されたテーブルは、そのインデックス・データを、ハッシュ化した名前を使用するグローバルに格納します。通常、テーブルに対してクエリを実行する場合、ハッシュ化した名前の方がパフォーマンスが高くなりますが、ユーザにとって理解しにくくなります。このパラメータが有効ではない場合、インデックス・データは、クラスに基づく名前が付いたグローバルに格納されます。

このパラメータの変更

ターミナルから DDLUseExtentSet に目的の値を設定するには、`%SYSTEM.SQL.Util` クラスの `SetOption` (“DDLUseExtentSet”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、`Config.SQL` クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、DDLUseExtentSet を変更できます。

DDLUseSequence

DDL 文によって作成されたテーブルで ID の割り当てに使用する関数を指定します。

構文

```
[SQL]      DDLUseSequence=n
```

n は 1 または 0 です。既定値は 1 です。

説明

DDLUseSequence が有効な場合 (n = 1)、DDL CREATE TABLE 文によって作成されたテーブルは ID の割り当てに \$SEQUENCE を使用します。このパラメータが有効ではない場合、テーブルは \$INCREMENT を使用します。

\$SEQUENCE は既定の関数で、より ID の割り当てに適しています。2 つの関数の比較は、“ObjectScript リファレンス”の“\$INCREMENT”の章にある“[\\$INCREMENT または \\$SEQUENCE](#)”のセクションを参照してください。

このパラメータの変更

ターミナルから DDLUseSequence に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“DDLUseSequence”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)”を参照)、DDLUseSequence を変更できます。

DefaultSchema

既定の SQL スキーマ名を設定します。

構文

```
[SQL]      DefaultSchema=n
```

n は、最大 128 文字の文字列です。既定文字列は、SQLUser です。

説明

DefaultSchema は、既定の SQL スキーマ名を定義します。既定のスキーマ名が使用されるのは、未修飾のテーブル名が SQL 文で出現し、`#import` 文が指定されていない場合です。この設定は、SQL スキーマ名とクラス・パッケージ名間のマッピングとは関係ありません。既定のスキーマ名を指定するだけです。

既定のスキーマ名として `_CURRENT_USER` を指定した場合、既定のスキーマ名は現在ログインしているプロセスのユーザ名になり、プロセスがログインしていない場合は、SQLUser が既定のスキーマ名になります。

既定のスキーマ名として `_CURRENT_USER/name` (`name` は任意に選択した文字列) を指定した場合、既定のスキーマ名は現在ログインしているプロセスのユーザ名になり、プロセスがログインしていない場合は、`name` が既定のスキーマ名として使用されます。例えば、プロセスがログインしていない場合、`_CURRENT_USER/HMO` では、HMO が既定のスキーマ名として使用されます。

詳細は、“[CREATE TABLE](#)” コマンドおよび “[CREATE VIEW](#)” コマンドを参照してください。

このパラメータの変更

管理ポータルの [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[既定のスキーマ] の設定に文字列を入力します。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、DefaultSchema を変更できます。

DelimitedIds

二重引用符で囲まれた文字列を、区切り識別子として解釈できるようにします。

構文

```
[SQL] DelimitedIds=n
```

n は 1 または 0 です。既定値は 1 です。

説明

DelimitedIds が有効な場合 (n = 1)、二重引用符で区切られた文字列 ("My String") は、SQL 文の区切り識別子と見なされます。このパラメータが有効ではない場合、二重引用符で区切られた文字列 ("My String") は、文字列定数またはリテラル文字列と見なされます。

詳細は、["SET OPTION" コマンド](#)を参照してください。[区切り識別子](#)の詳細は、“識別子”の章を参照してください。

このパラメータの変更

ターミナルから DelimitedIds に目的の値を設定するには、`%SYSTEM.SQL.Util` クラスの `SetOption("DelimitedIdentifiers")` メソッドを使用します。詳細は、[クラスリファレンス](#)を参照してください。

あるいは、`Config.SQL` クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで CPF を編集することでも ("[アクティブな CPF の編集](#)" を参照)、DelimitedIds を変更できます。

DropDelete

DROP TABLE でテーブルに加えテーブルのデータも削除するかどうかを指定します。

構文

```
[SQL]      DropDelete=n
```

n は 1 または 0 です。既定値は 1 です。

説明

DropDelete が有効な場合 (n = 1)、DROP TABLE 文を実行すると、テーブルおよびそのデータが削除されます。このパラメータが有効ではない場合、DROP TABLE 文を実行すると、テーブルは削除されますがデータは削除されません。

詳細は、“[DROP TABLE](#)” コマンドを参照してください。

このパラメータの変更

ターミナルから DropDelete に目的の値を設定するには、`%SYSTEM.SQL.Util` クラスの `SetOption` (“DDLDropTabDel-Data”) メソッドを使用します。詳細は、[クラスリファレンス](#)を参照してください。

あるいは、`Config.SQL` クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)”を参照)、DropDelete を変更できます。

ECPSync

クライアントとサーバのキャッシュを同期します。

構文

```
[SQL]      ECPSync=n
```

n は 1 または 0 です。既定値は 0 です。

説明

ECPSync が有効な場合 (n = 1)、SELECT 文が実行されるたびに、InterSystems IRIS® Data Platform は、データベース・サーバに対する保留中の ECP (エンタープライズ・キャッシュ・プロトコル) 要求すべてを強制的に実行します。これが完了すると、クライアント・キャッシュの同期が保証されます。

ECP は分散データ・キャッシュ・アーキテクチャであり、異種のサーバ・システムで構成されるネットワークに分散されるデータとロックを管理します。詳細は、“データベースの問い合わせ” の章の “[クエリと ECP](#)” を参照してください。

このパラメータの変更

InterSystems ターミナルから ECPSync に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“ECPSync”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、ECPSync を変更できます。

ExtrinsicFunctions

SQL 文で外部関数を使用可能にします。

構文

```
[SQL]      ExtrinsicFunctions=n
```

n は 1 または 0 です。既定値は 0 です。

説明

ExtrinsicFunctions が有効な場合 (n = 1)、ODBC、JDBC、およびダイナミック・クエリを介して外部関数を SQL 文で使用できます。

詳細は、“[SELECT](#)” コマンドを参照してください。

このパラメータの変更

ターミナルから ExtrinsicFunctions に目的の値を設定するには、`%SYSTEM.SQL.Util` クラスの `SetOption`(“AllowExtrinsicFunctions”) メソッドを使用します。詳細は、[クラスリファレンス](#)を参照してください。

あるいは、`Config.SQL` クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)”を参照)、ExtrinsicFunctions を変更できます。

FastDistinct

SQL DISTINCT の最適化を許可します。

構文

```
[SQL]      FastDistinct=n
```

n は 1 または 0 です。既定値は 1 です。

説明

FastDistinct が有効な場合 (n = 1)、インデックスの効果的な使用により (可能な場合)、DISTINCT または GROUP BY が含まれる SQL クエリを実行するときの効率が向上します。

注意 このクエリによって返される値は、インデックス内に格納されるときと同じ方法で照合されます。このため、このクエリの結果はすべて大文字になることがあります。これは、大文字と小文字を区別するアプリケーションに影響する場合があります。

詳細は、SELECT 文の “[GROUP BY](#)” 節および “[DISTINCT](#)” 節を参照してください。

このパラメータの変更

FastDistinct は既定で有効になっています。管理ポータルの [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[GROUP BY および DISTINCT クエリで元の値を生成する] を選択して、FastDistinct を無効にします。

ターミナルから FastDistinct に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“FastDistinct”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、FastDistinct を変更できます。

IdKey

主キー制約の振る舞いを設定します。

構文

```
[SQL]      IdKey=n
```

n は 1 または 0 です。既定値は 1 です。

説明

IdKey が有効な場合 (n = 1)、DDL を介して主キー制約を指定すると、その主キー制約はクラス定義の IDKey インデックスになりません。

このパラメータが有効ではない場合、DDL を介して指定された主キー制約は、クラス定義の IDKey インデックスになります。このオプションを使用すると、通常はパフォーマンスが向上しますが、主キーのフィールドを更新できなくなります。

詳細は、“[SET OPTION](#)” コマンド、“[CREATE TABLE](#)” コマンド、および “[ALTER TABLE](#)” コマンドを参照してください。

このパラメータの変更

管理ポータルの [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[DDL を使用して作成されたテーブルの ID キーとして主キーを定義する] を選択して、IdKey を無効にします。

ターミナルから IdKey に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“DDLPrimaryKeyNotIDKey”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、IdKey を変更できます。

IdTrxFrom

識別子変換の文字の“変換元”リストを定義します。

構文

```
[SQL]      IdTrxFrom=n
```

n は、最大 256 文字の文字列です。既定の文字列は、~ `@#\$\$%^&*()_+--[]\{|;':",./<>?". です。

説明

IdTrxFrom は、DDL 識別子変換マッピングの“変換元”リストを指定する文字列です。これらのマッピングによって、SQL 識別子をオブジェクト識別子に変換する場合に、有効な SQL 識別子文字のフィルタ処理や変更が行われます。DDL 実行時に SQL 識別子をオブジェクト識別子に変換する場合、“変換元”文字列の文字は、“変換先”文字列の文字に変換されます。

詳細は、“[識別子](#)”の章を参照してください。パラメータ [IdTrxTo](#) も参照してください。

このパラメータの変更

ターミナルから IdTrxFrom に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetDDLIdentifierTranslations() メソッドを使用します。詳細は、[クラスリファレンス](#)を参照してください。

あるいは、Config.SQL クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)”を参照)、IdTrxFrom を変更できます。

IdTrxTo

識別子変換の文字の“変換先”リストを定義します。

構文

```
[SQL]      IdTrxTo=n
```

n は、最大 256 文字の文字列です。既定値は空文字列です。

説明

IdTrxTo は、DDL 識別子変換マッピングの“変換先”リストを指定する文字列です。

詳細は、“[識別子](#)”の章を参照してください。“[IdTrxFrom](#)”も参照してください。

このパラメータの変更

ターミナルから IdTrxTo に目的の値を設定するには、`%SYSTEM.SQL.Util` クラスの `SetDDLIdentifierTranslations()` メソッドを使用します。詳細は、[クラスリファレンス](#)を参照してください。

あるいは、`Config.SQL` クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)”を参照)、IdTrxTo を変更できます。

LockThreshold

SQL テーブルレベルのロックしきい値を設定します。

構文

```
[SQL]      LockThreshold=n
```

n は負ではない任意の整数です。既定値は 1000 です。

説明

LockThreshold パラメータは、自動ロック・エスカレーションしきい値です。1 つのトランザクション内で 1 つのテーブルに対して実行された挿入、更新、および削除の数がこの数に達すると、テーブルレベル・ロックがトリガされます。

以下の例を見てみましょう。あるプロセスが、2000 行を挿入するトランザクションを開始します。LockThreshold は 1000 に設定されています。1001 番目の行を挿入した後、このプロセスは、引き続き個々の行をロックするのではなく、テーブルレベルのロックを取得しようとします。これにより、ロックの合計数を減らし、ロック・テーブルが一杯になるのを防ぎます。

自動ロック・エスカレーションは、ロック・テーブルのオーバーフローを防ぐことを目的としています。詳細は、“InterSystems SQL リファレンス”の“INSERT”のリファレンス・ページにある[“トランザクションのロックしきい値の変更”](#)のセクションを参照してください。

このパラメータの変更

管理ポータルの [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[ロック・エスカレーションしきい値] フィールドに数値を入力します。

ターミナルから LockThreshold に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“LockThreshold”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも ([“アクティブな CPF の編集”](#) を参照)、LockThreshold を変更できます。

関連項目

- [gmheap](#) パラメータおよび [locksiz](#) パラメータ (ロック・テーブルのサイズを増やすため)。
- [ロック・テーブル](#)
- “InterSystems SQL リファレンス”の[“トランザクションのロックしきい値の変更”](#)。

LockTimeout

SQL ロック・タイムアウトを設定します。

構文

```
[SQL]      LockTimeout=n
```

n は、0 から 32,767 の範囲の整数です。既定値は 10 です。

説明

LockTimeout は、SQL 文の実行時に実行される InterSystems IRIS® Data Platform ロックのロック・タイムアウト時間 (秒) です。最大値は 32,767 秒 (9 時間) です。

詳細は、“[SET OPTION](#)” コマンドを参照してください。

このパラメータの変更

管理ポータル の [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[ロックタイムアウト (秒)] 設定に数値を入力します。

InterSystems ターミナルから LockTimeout に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“Lock-Timeout”) メソッドを使用します。詳細は、[クラスリファレンス](#)を参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、LockTimeout を変更できます。

ODBCVarcharMaxlen

VarChar 型の ODBC フィールドの最大長 (MaxLen) を設定します。

構文

```
[SQL]      ODBCVarcharMaxlen=n
```

n は負ではない任意の整数です。最大値は、[最大文字列長](#)です。既定値は 4096 です。

説明

ODBCVarcharMaxlen は、InterSystems IRIS® Data Platform から ODBC に通知される MaxLen (最大長) で、データ型が VarChar のフィールドです。

このパラメータの変更

Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、ODBCVarcharMaxlen を変更できます。

ParameterSampling

パラメータ収集が有効かどうかを指定します。

構文

```
[SQL]      ParameterSampling=n
```

n は 1 または 0 です。既定値は 0 です。

説明

ParameterSampling が有効 (n = 1) な場合、文を作成するときに InterSystems SQL によってクエリ・パラメータの完全なセットが保存されます。このデータは、INFORMATION_SCHEMA.STATEMENT_PARAMETER_STATS に投影されます。このデータには、クエリの効率化に関する情報が記述されています。パラメータ収集の有効化はスキーマの効果の評価 (特に新規スキーマをテストするとき) に役立ちます。

このパラメータの変更

管理ポータルの [SQL] ページ ([システム管理] > [構成] > [SQL とオブジェクトの設定] > [SQL]) で [クエリ実行のパラメータを収集するパラメータ収集を有効にする] を選択して ParameterSampling を有効にします。

InterSystems ターミナルから ParameterSampling に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption("ParameterSampling") メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも ("[アクティブな CPF の編集](#)" を参照)、ParameterSampling を変更できます。

QueryProcedures

すべてのクラス・クエリを SQL ストアド・プロシージャとして投影するかどうかを指定します。

構文

```
[SQL]      QueryProcedures=n
```

n は 1 または 0 です。既定値は 0 です。

説明

QueryProcedures が有効な場合 (n = 1)、クエリの SqlProc 値に関係なく、すべての SQL クラス・クエリが SQL ストアド・プロシージャとして投影されます。このパラメータが有効ではない場合、SqlProc=1 で定義されているクラス・クエリのみがストアド・プロシージャとして投影されます。

この設定を変更した場合、この変更を有効にするには、クラス・クエリを含めてクラスを再コンパイルする必要があります。CPF でこの設定を変更しても、変更内容を有効化するためにインスタンスを再起動する必要はありません。

このパラメータの変更

ターミナルから QueryProcedures に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“QueryProcedures”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも ([“アクティブな CPF の編集”](#) を参照)、QueryProcedures を変更できます。

RTPC

実行時プラン選択 (RTPC) によるクエリ最適化を有効にします。

構文

```
[SQL]      RTPC=n
```

n は 1 または 0 です。既定値は 1 です。

説明

RTPC が有効 (*n* = 1) な場合、InterSystems SQL ではクエリ入力値に基づいてクエリが幅広く最適化されます。例えば、RTPC は以下を実行します。

- ・ 異常値スキャンを実行し、異常値の情報に基づいてクエリを最適化します。
- ・ 詳細なテーブル統計に基づき、範囲条件の選択性を効率的に評価します。
- ・ 明示的および暗黙的な真理値条件を評価します。
- ・ リスト条件の選択性を効率的に評価します。

RTPC クエリ最適化の詳細は、“[実行時プラン選択の使用法](#)” を参照してください。

RTPC と [BiasQueriesAsOutlier](#) の両方を同時に有効にすることはできません。RTPC と [BiasQueriesAsOutlier](#) の両方を 1 に設定すると RTPC が有効になり、[BiasQueriesAsOutlier](#) 設定は無視されます。RTPC が設定されている場合、InterSystems SQL は、クエリごとに異常値の最適化を使用できます。

[AdaptiveMode](#) が有効 (1 に設定) で RTPC が無効な場合、アダプティブ・モードによって RTPC の設定がオーバーライドされ、実行時プラン選択が有効になります。

このパラメータの変更

ターミナルから RTPC に目的の値を設定するには、`%SYSTEM.SQL.Util` クラスの `SetOption(“RTPC”)` メソッドを使用します。詳細は、[クラスリファレンス](#)を参照してください。

あるいは、`Config.SQL` クラス ([クラス・リファレンス](#)を参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)”を参照)、RTPC を変更できます。

ReferentialChecks

外部キー制約の検証を有効にします。

構文

```
[SQL]      ReferentialChecks=n
```

n は 1 または 0 です。既定値は 1 です。

説明

ReferentialChecks が有効な場合 (n = 1)、InterSystems IRIS® Data Platform により、INSERT、UPDATE、DELETE、および TRUNCATE TABLE の各操作の外部キー制約が検証されます。このパラメータが有効ではない場合、InterSystems IRIS では外部キー制約の検証は行われません。

詳細は、“InterSystems SQL リファレンス” の “[DELETE](#)” コマンド、“[INSERT](#)” コマンド、“[TRUNCATE TABLE](#)” コマンド、および “[UPDATE](#)” コマンドを参照してください。

このパラメータの変更

InterSystems ターミナルから ReferentialChecks に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“FilerRefIntegrity”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、ReferentialChecks を変更できます。

SaveMAC

クエリ・キャッシュのルーチンのソース・コードを保存します。

構文

```
[SQL]      SaveMAC=n
```

n は 1 または 0 です。既定値は 0 です。

説明

SaveMac が有効な場合 (*n* = 1)、ダイナミック SQL を介して作成されたクエリ・キャッシュのルーチンのソース・コード (.MAC と .INT) が保存されます。

このパラメータの変更

管理ポータルの [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[クエリ・キャッシュ・ソースの保持] を選択して SaveMac を有効にします。

ターミナルから SaveMac に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“CachedQuerySaveSource”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、SaveMac を変更できます。

TCPKeepAlive

キープ・アライブ・メッセージ間の秒数を設定します。

構文

```
[SQL]      TCPKeepAlive=n
```

n は、30 から 432,000 の範囲の整数です。既定値は 300 です。

説明

TCPKeepAlive は、キープ・アライブ・メッセージ間の秒数です。この設定は、Windows および Linux で実行されている InterSystems IRIS® Data Platform にのみ適用されます。既定値は 300 秒 (5 分) で、最大値は 432,000 (5 日) です。この値が 0 の場合、オペレーティング・システムの既定値が使用されます。

詳細は、“[TCP クライアント/サーバ通信](#)” の章を参照してください。

このパラメータの変更

管理ポータルの [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[クライアント接続の TCP キープアライブ (秒)] 設定に値を入力します。

InterSystems ターミナルから TCPKeepAlive に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“TCP-KeepAlive”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、TCPKeepAlive を変更できます。

TCP デバイスの場合、OPEN コマンドまたは USE コマンドを使用してこの間隔を指定することもできます。8 番目のパラメータを指定するか (例 : 0 tcp:("SERVER":port:mode:::::keepalive)、キーワード /KEEPALIVE=n を指定します。TCP デバイスの OPEN コマンド・パラメータは

hostname{ :port{ :mode{ :terminators{ :ibfsz{ :obfsz{ :queuesize{ :keepalivetime{ }}}}}}} です。

OPEN および USE のコマンドと引数の詳細は、例を含め、“[ObjectScript リファレンス](#)” を参照してください。

TODATEDefaultFormat

SQL TO_DATE() 関数の既定の日付書式を設定します。

構文

```
[SQL]      TODATEDefaultFormat=n
```

n は、TO_DATE() 関数に対応する書式の任意の文字列です。既定の文字列は、DD MON YYYY です。

説明

値 n は、書式の指定なしに SQL TO_DATE() が呼び出されたときに、この関数を使用する書式の文字列を指定します。

例

以下に例を示します。

```
TODATEDefaultFormat=DD MON YYYY
```

以下に別の例を示します。

```
TODATEDefaultFormat=YYYY DD MM
```

有効な日付文字列の他の例と詳細は、TO_DATE のリファレンス・ページにある [format](#) 引数の説明を参照してください。

このパラメータの変更

管理ポータル の [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[TO_DATE の既定の形式] フィールドに有効な形式の文字列を入力します。

ターミナルから TODATEDefaultFormat に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“ToDateDefaultFormat”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することでも (“[アクティブな CPF の編集](#)” を参照)、TODATEDefaultFormat を変更できます。

TimePrecision

SQL スカラ時刻関数の既定の時刻精度を設定します。

構文

```
[SQL]      TimePrecision=n
```

n は、0 から 9 の範囲の整数です。既定値は 0 です。

説明

TimePrecision は、SQL スカラ関数 GETDATE()、CURRENT_TIME、CURRENT_TIMESTAMP、GETUTCDATE、および UNIX_TIMESTAMP によって返される値の小数点以下の既定桁数を定義します。これらの関数で返される値の秒の小数部の桁数は、小数点以下 n 桁です。実際に使用可能な精度はプラットフォームに依存し、システムで使用可能な精度を超えた精度の桁はゼロとして返されます。

このパラメータの変更

管理ポータルの [SQL] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [SQL]) で、[GETDATE()、CURRENT_TIME、CURRENT_TIMESTAMP のデフォルトの時刻精度] の設定で、小数点以下の桁数を選択します。

ターミナルから TimePrecision に目的の値を設定するには、%SYSTEM.SQL.Util クラスの SetOption(“DefaultTime-Precision”) メソッドを使用します。詳細は、クラスリファレンスを参照してください。

あるいは、Config.SQL クラス (クラス・リファレンスを参照) で、テキスト・エディタで CPF を編集することで (“[アクティブな CPF の編集](#)” を参照)、または SET OPTION コマンドを使用することでも、TimePrecision を変更できます。

[SqlSysDatatypes]

このトピックでは、CPF の [SqlSysDatatypes] セクションのパラメータについて説明します。

システム・データ型

SQL データ型を同等の InterSystems IRIS® Data Platform データ型にマッピングします。

構文

```
[SqlSysDatatypes]      x=a
```

x は、SQL データ型の名前です。a は、同等の InterSystems IRIS データ型です。

説明

[SqlSysDatatypes] セクションには、システム定義のデータ型の記述が含まれます。以下のように、それぞれの記述が、x=a の形式で 1 つの SQL データ型を同等の InterSystems IRIS データ型にマッピングします。

- ・ 各キーワード x は、SQL データ型の名前と使用可能な引数です。
- ・ 値 a は、同等の InterSystems IRIS データ型で、引数に制約がある場合はそれも含まれます。

例

構成パラメータ・ファイルから抜粋した以下の内容は、[SqlSysDatatypes] の B という文字で始まる項目を示しています。システム定義 DDL データ型のすべてのマッピングのテーブルは、“[DDL データ型のテーブル](#)”を参照してください。

```
[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
```

このパラメータの変更

管理ポータル内の [システム定義 DDL マッピング] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [システム定義 DDL マッピング]) で、[編集] を選択して、データ型の定義を変更します。

管理ポータルを使用する代わりに、テキスト・エディタで CPF を編集することで (“[アクティブな CPF の編集](#)”を参照)、データ型の定義を変更できます。

関連項目

- ・ [ユーザ・データ型](#)
- ・ [データ型](#)

[SqlUserDatatypes]

このトピックでは、CPF の [SqlUserDatatypes] セクションのパラメータについて説明します。

ユーザ・データ型

SQL データ型を同等の InterSystems IRIS® Data Platform データ型にマッピングします。

構文

```
[SqlUserDatatypes]      x=a
```

x は、SQL データ型の名前です。a は、同等の InterSystems IRIS データ型です。

説明

[SqlUserDatatypes] セクションには、ユーザ定義のデータ型の記述が含まれます。以下のように、それぞれの記述が、x=a の形式で 1 つの SQL データ型を同等の InterSystems IRIS データ型にマッピングします。

- ・ 各キーワード x は、SQL データ型の名前と使用可能な引数です。
- ・ 値 a は、同等の InterSystems IRIS データ型で、引数に制約がある場合はそれも含まれます。

例

CPF の以下の行で、文に MYVARCHAR(10) と記述すると、%Library.String(MAXLEN=10,TRUNCATE=0) 型のプロパティが作成されます。

```
[SqlUserDatatypes]
MYVARCHAR(%1)=%Library.String(MAXLEN=%1,TRUNCATE=0)
```

このパラメータの変更

管理ポータル の [ユーザ定義 DDL マッピング] ページ ([システム管理] > [構成] > [SQL およびオブジェクトの設定] > [ユーザ DDL マッピング]) で、[新規ユーザ定義 DDL マッピング作成] をクリックして新しいデータ型マッピングを追加するか、[編集] をクリックして既存のデータ型マッピングを変更します。

管理ポータルを使用する代わりに、Config.SqlUserDatatypes クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、データ型の定義を変更できます。

関連項目

- ・ [システム・データ型](#)
- ・ [データ型](#)

[Startup]

このトピックでは、CPF の [Startup] セクションにある **【開始設定】** のパラメータについて説明します。これらのパラメータの多くは、ミラー関連のパラメータやシャード・クラスタ関連のパラメータを含め、構成マージを使用する導入に含まれます。

CallinHalt

コールインのクローズ時にカスタム・ルーチンを実行できるようにします。

構文

```
[Startup]      CallinHalt=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

CallinHalt が有効な場合 (n = 1)、InterSystems IRIS® Data Platform は、外部プログラムが CALLIN を終了するたびに、CALLIN^%ZSTOP ルーチン・エントリを実行します。このパラメータが有効ではない場合、ルーチンは実行されません。

CALLIN の詳細は、“[コールイン API の使用法](#)” を参照してください。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[CallinHalt]** の行の **[編集]** を選択します。**[CallinHalt]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、CallinHalt を変更できます。

関連項目

[ProcessStart](#) および [ProcessHalt](#) は、フォアグラウンド・プロセスに対応します。これらは、ターミナル・セッションを介して起動されるか、フォアグラウンドで実行するために設定されるプロセスです。

[JobStart](#) および [JobHalt](#) は、バックグラウンド・プロセスに対応します。これには、JOB コマンドを介して起動されるプロセスと、ODBC、任意の言語バインディングなどのバックグラウンド・サーバ・プロセスとが含まれます。

[SystemStart](#) および [SystemHalt](#) は、InterSystems IRIS インスタンスの開始またはシャットダウンに対応します。

[CallinStart](#) および [CallinHalt](#) は、CALLIN を実行する外部プログラムに対応します。

これらすべてのパラメータの詳細は、“[%ZSTART ルーチンと %ZSTOP ルーチンによる開始動作と停止動作のカスタマイズ](#)” を参照してください。

CallinStart

コールインの初期化時にカスタム・ルーチンを実行できるようにします。

構文

```
[Startup]      CallinStart=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

CallinStart が有効な場合 (n = 1)、InterSystems IRIS® Data Platform は、外部プログラムが CALLIN を開始するたびに、CALLIN^%ZSTART ルーチン・エントリを実行します。このパラメータが有効ではない場合、ルーチンは実行されません。

CALLIN の詳細は、“[コールイン API の使用法](#)” を参照してください。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[CallinStart]** の行の **[編集]** を選択します。**[CallinStart]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、CallinStart を変更できます。

関連項目

[ProcessStart](#) および [ProcessHalt](#) は、フォアグラウンド・プロセスに対応します。これらは、ターミナル・セッションを介して起動されるか、フォアグラウンドで実行するために設定されるプロセスです。

[JobStart](#) および [JobHalt](#) は、バックグラウンド・プロセスに対応します。これには、JOB コマンドを介して起動されるプロセスと、ODBC、任意の言語バインディングなどのバックグラウンド・サーバ・プロセスとが含まれます。

[SystemStart](#) および [SystemHalt](#) は、InterSystems IRIS インスタンスの開始またはシャットダウンに対応します。

CallinStart および [CallinHalt](#) は、CALLIN を実行する外部プログラムに対応します。

これらすべてのパラメータの詳細は、“[%ZSTART ルーチンと %ZSTOP ルーチンによる開始動作と停止動作のカスタマイズ](#)” を参照してください。

CliSysName

ローカル・システムのノード名を設定します。

構文

```
[Startup]      CliSysName=n
```

n は、最大 64 文字の文字列です。既定値は空文字列です。

説明

CliSysName は、このコンピュータのノード名で、以下として使用されます。

- ・ サーバがクライアントを識別できるように、ECP のネットワーク・サーバに送信されるノード名。
- ・ 一意の **\$JOB** 値のノード名。この値は、\$JOB を使用して、ネットワーク接続された複数のシステムからアクセスされるグローバルのインデックスを作成する際に役立ちます。
- ・ 特定の形式の **\$SYSTEM** 関数によって返されるノード名に、InterSystems IRIS® Data Platform インスタンス名を連結したもの (nodename:instancename)。この連結された文字列は、監査ファイルに記録されます。

名前を指定しない場合、InterSystems IRIS はコンピュータの設定を読み取り、そのコンピュータの“ホスト名”をクライアントのノード名として使用します。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[CliSysName]** の行の **[編集]** を選択します。目的のノード名を入力します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、CliSysName を変更できます。

DBSizesAllowed

データベースの作成時に選択できるデータベース・ブロック・サイズを指定します。

構文

```
[Startup]    DBSizesAllowed=n[n,n...]
```

n には、8192、16384、32768、または 65536 を指定できます。既定値は 8192 です。

説明

DBSizesAllowed には、データベースの作成時に選択できるデータベース・ブロック・サイズ (バイト) を記述します。

データベースの作成および管理の詳細は、“[データベースの構成](#)” を参照してください。

例

```
DBSizesAllowed=8192,16384
```

```
DBSizesAllowed=8192,65536
```

このパラメータの変更

管理ポータルの [開始] ページ ([システム管理] > [構成] > [追加設定] > [開始]) で、[DBSizesAllowed] の行の [編集] を選択します。目的の各データベース・ブロック・サイズを選択します。8192 のチェックを外すことはできません。

重要 追加のデータベース・ブロック・サイズを有効にする場合、[globals](#) パラメータを使用して、そのブロック・サイズにメモリを割り当てる必要があります。これにより、InterSystems IRIS® Data Platform は、そのサイズに必要なグローバル・バッファのプールを作成できます。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、DBSizesAllowed を変更できます。

関連項目

- このリファレンスの “[Config]” セクションの “[globals](#)” パラメータ
- [データベースの構成](#)
- [データベース・キャッシュおよびルーチン・キャッシュへのメモリの割り当て](#)
- [ラージ・ブロック・サイズに関する考慮事項](#)

DefaultPort

InterSystems IRIS® Data Platform スーパーサーバのポート番号を設定します。

構文

```
[Startup]      DefaultPort=n
```

n は有効なポート番号です。既定値は 1972 です。

説明

DefaultPort は、InterSystems IRIS スーパーサーバのポート番号です。スーパーサーバは、指定したポート (既定では 1972) で InterSystems IRIS への着信接続をリッスンし、それを適切なサブシステムに配信します。

標準の InterSystems IRIS インストールでは、スーパーサーバ・ポート番号として 1972 が設定されます。そのポートが同じシステム上の別の InterSystems IRIS インスタンスによって使用されている場合、この値は 51773 またはこれ以降の次に使用可能な番号に設定されます。

このパラメータの変更

管理ポータル の **[メモリと開始設定]** ページ (**[システム管理]** > **[構成]** > **[システム構成]** > **[メモリと開始設定]**) で、**[スーパーサーバポート番号]** フィールドに数値を入力します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、DefaultPort を変更できます。

DefaultPortBindAddress

InterSystems IRIS® Data Platform スーパーサーバをバインドする IP アドレスを設定します。

構文

```
[Startup]      DefaultPortBindAddress=nnn.nnn.nn.nn
```

nnn.nnn.nn.nn は有効な IP アドレスです。既定では、何も指定されていません。

説明

DefaultPortBindAddress は、スーパーサーバをバインドする、ホスト・システム上の IP アドレスです。スーパーサーバは、ODBC、JDBC、およびその他の接続テクノロジーのクライアント接続を許可するプロセスです。ホストの他の IP アドレスからのスーパーサーバ・ポートへの要求は、許可されません。これにより、スーパーサーバへの接続をマルチホーム・ホストの 1 つのアドレスに制限できます。

このプロパティを設定しないと、スーパーサーバはホスト上のすべての IP アドレスからの要求を許可します。既定では、すべてのアドレスを許可します。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[DefaultPortBindAddress]** の行の **[編集]** を選択します。IP アドレスを入力します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、DefaultPortBindAddress を変更できます。

EnableSharding

このインスタンスに対してシャーディング・サービスを有効にします。

構文

```
[Startup]      EnableSharding=n
```

n は、1 (真) または 0 (偽) のいずれかです。

説明

このパラメータが有効な場合、(n = 1)、InterSystems IRIS® データ・プラットフォームは、インスタンスの開始時にシャーディング・サービス (**%Service_Sharding**) を有効にします。EnableSharding が無効な場合 (n = 0)、インスタンスの開始時に **%Service_Sharding** は変更されません。つまり、EnableSharding は、シャーディング・サービスを有効にする目的でのみ使用でき、無効にする目的では使用できません。

EnableSharding は、既定の CPF にはありませんが、[構成マージ・ファイル](#)にパラメータとして含めることができます。EnableSharding パラメータを使用せずに直接 **%Service_Sharding** を有効にするには、管理ポータルの **[サービス]** ページ (**[システム管理]** > **[セキュリティ]** > **[サービス]**) を使用します。

このパラメータの変更

EnableSharding は、既定の CPF にはありません。テキスト・エディタを使用して、CPF の **[Startup]** セクションにこのパラメータを手動で追加できます ("[アクティブな CPF の編集](#)" を参照)。

EnableSharding を追加して InterSystems IRIS を再起動した後、テキスト・エディタ、Config.Startup クラス (クラス・リファレンスを参照)、または管理ポータル **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) を使用して、このパラメータを変更できます。

関連項目

- ・ ["サービス"](#)

EnableVSSBackup

ボリューム・シャドウ・コピー・サービス (VSS) によるバックアップを許可します。Windows システムでのみ有効です。

構文

```
[Startup]      EnableVSSBackup=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

EnableVSSBackup が有効な場合 (n = 1)、InterSystems IRIS® Data Platform は Windows で VSS をサポートします。VSS は Windows でのみ使用可能です。他のプラットフォームでは、EnableVSSBackup パラメータは無視されます。

VSS またはその他の方法でバックアップを作成する方法については、“[バックアップの方法](#)” を参照してください。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[EnableVSSBackup]** の行の **[編集]** を選択します。**[EnableVSSBackup]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、EnableVSSBackup を変更できます。

EnsembleAutoStart

InterSystems IRIS® Data Platform の起動時にプロダクションを自動開始できるようにします。

構文

```
[Startup]      EnsembleAutoStart=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

EnsembleAutoStart が有効な場合、InterSystems IRIS を開始すると、各相互運用対応ネームスペースで自動開始するように設定したプロダクションが開始します。問題のあるプロダクションに関するデバッグ状況を簡易化するには、この設定を False にすることで、プロダクションが起動することを防止できます。

この設定がプロダクション設定とどのように関係して機能するのかの詳細は、“プロダクションの開始と停止” で、[プロダクション自動開始] フィールドの説明を参照してください。

このパラメータの変更

管理ポータル の **[開始]** ページ ([システム管理] > [構成] > [追加設定] > [開始]) で、[EnsembleAutoStart] の行の **[編集]** を選択します。[EnsembleAutoStart] を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、EnsembleAutoStart を変更できます。

ErrorPurge

エラー・グローバルを保存する日数を設定します。

構文

```
[Startup]      ErrorPurge=n
```

n は、1 から 1000 の範囲の整数です。既定値は 30 です。

説明

ErrorPurge は、[%ETN](#) エラー・ハンドラのエラー・グローバルを保存する日数です。この日数を超えたエラーは、次の InterSystems IRIS® Data Platform の再起動時に削除されます。

このパラメータの変更

管理ポータル[の](#) **【開始】** ページ (**【システム管理】** > **【構成】** > **【追加設定】** > **【開始】**) で、**[ErrorPurge]** の行の **【編集】** を選択します。日数を入力します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、ErrorPurge を変更できます。

この設定を編集した場合、次回 InterSystems IRIS を再起動したときに、変更内容が適用されます。

FIPSMODE

Red Hat Linux で FIPS 140-2 に準拠したデータベース暗号化用ライブラリを有効にします。

構文

```
[Startup]    FIPSMODE=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

FIPSMODE が有効な場合、InterSystems IRIS® Data Platform は、Red Hat Enterprise Linux 6.6 (またはそれ以降のマイナー・バージョン) および Red Hat Enterprise Linux 7.1 (またはそれ以降のマイナー・バージョン) (x86-64) で FIPS 140-2 に準拠したデータベース暗号化用ライブラリを使用します。

注釈 FIPSMODE の有効化は、暗号化データベースにのみ影響します。データベースを暗号化するには、[データベースの作成プロセス](#)中に [\[データベース暗号化?\]](#) オプションを選択します。

詳細は、"[データベース暗号化の FIPS 140-2 準拠](#)" の記事を参照してください。

このパラメータの変更

管理ポータル の [\[開始\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[開始\]](#)) で、[\[FIPSMODE\]](#) の行の [\[編集\]](#) を選択します。FIPS 140-2 に準拠したデータベース暗号化用ライブラリを使用する場合は、[\[FIPSMODE\]](#) を選択します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、FIPSMODE を変更できます。

IPv6

InterSystems IRIS® Data Platform が IPv6 アドレスを受け入れることができるようにします。

構文

```
[Startup]      IPv6=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 0 です。

説明

IPv6 は、インスタンスを IPv6 (Internet Protocol Version 6) アドレスを使用する IPv6 ネットワークで運用するかどうかを制御します。詳細は、“[IPv6 のサポート](#)” を参照してください。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[IPv6]** の行の **[編集]** を選択します。**[IPv6]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、IPv6 を変更できます。

JobHalt

バックグラウンド・プロセス (ジョブ) のシャットダウン中にカスタム・ルーチンを実行できるようにします。

構文

```
[Startup]      JobHalt=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

JobHalt が有効な場合 (n = 1)、InterSystems IRIS® Data Platform は、バックグラウンド・プロセスの終了時に JOB^%ZSTOP ルーチン・エントリを実行します。バックグラウンド・プロセスには、JOB コマンドを介して起動されるプロセスと、ODBC、任意の言語の結合などのバックグラウンド・サーバ・プロセスとが含まれます。このパラメータが有効ではない場合、ルーチンは実行されません。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[JobHalt]** の行の **[編集]** を選択します。**[JobHalt]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、JobHalt を変更できます。

関連項目

[ProcessStart](#) および [ProcessHalt](#) は、フォアグラウンド・プロセスに対応します。これらは、ターミナル・セッションを介して起動されるか、フォアグラウンドで実行するために設定されるプロセスです。

[JobStart](#) および [JobHalt](#) は、バックグラウンド・プロセスに対応します。これには、JOB コマンドを介して起動されるプロセスと、ODBC、任意の言語・バインディングなどのバックグラウンド・サーバ・プロセスとが含まれます。

[SystemStart](#) および [SystemHalt](#) は、InterSystems IRIS インスタンスの開始またはシャットダウンに対応します。

[CallinStart](#) および [CallinHalt](#) は、CALLIN を実行する外部プログラムに対応します。

これらすべてのパラメータの詳細は、"[%ZSTART ルーチンと %ZSTOP ルーチンによる開始動作と停止動作のカスタマイズ](#)" を参照してください。

JobServers

JOB サーバの数を入力します。

構文

```
[Startup]      JobServers=n
```

n は、0 から 2000 の範囲の整数です。既定値は 0 です。

説明

JobServers は、InterSystems IRIS® Data Platform によって起動する JOB サーバの数です。

多数の JOB サーバを実行すると、大量のメモリとプロセスを消費しますが、InterSystems IRIS ではシステム・レベルでプロセスを起動して初期化する必要がないため、ジョブ起動プロセスを高速化できます。

JOB サーバを使用するのが最も適しているのは、Job コマンドによってアプリケーションで一時的なプロセスが多数作成される場合です。このようなプロセスでは、オペレーティング・システムのプロセス作成オーバーヘッドがプロセスの実行コスト全体の大半を占めるため、JOB サーバを使用した方が効果的な場合があります。バックグラウンド・プロセスで大きなタスクを実行する場合は、JOB サーバを使用してもあまり効果はありません。

このパラメータの変更

管理ポータルの **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[JobServers]** の行の **[編集]** を選択します。JOB サーバの数を入力します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、JobServers を変更できます。

JobStart

バックグラウンド・プロセス (ジョブ) の開始中にカスタム・ルーチンを実行できるようにします。

構文

```
[Startup]      JobStart=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

JobStart が有効な場合 (n = 1)、InterSystems IRIS® Data Platform は、バックグラウンド・プロセスの開始時に JOB`%ZSTART ルーチン・エントリを実行します。バックグラウンド・プロセスには、JOB コマンドを介して起動されるプロセスと、ODBC、任意の言語の結合などのバックグラウンド・サーバ・プロセスとが含まれます。このパラメータが有効ではない場合、ルーチンは実行されません。

このパラメータの変更

管理ポータルの **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[JobStart]** の行の **[編集]** を選択します。**[JobStart]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、JobStart を変更できます。

関連項目

[ProcessStart](#) および [ProcessHalt](#) は、フォアグラウンド・プロセスに対応します。これらは、ターミナル・セッションを介して起動されるか、フォアグラウンドで実行するために設定されるプロセスです。

JobStart および [JobHalt](#) は、バックグラウンド・プロセスに対応します。これには、JOB コマンドを介して起動されるプロセスと、ODBC、任意の言語・バインディングなどのバックグラウンド・サーバ・プロセスとが含まれます。

[SystemStart](#) および [SystemHalt](#) は、InterSystems IRIS インスタンスの開始またはシャットダウンに対応します。

[CallinStart](#) および [CallinHalt](#) は、CALLIN を実行する外部プログラムに対応します。

これらすべてのパラメータの詳細は、"[%ZSTART ルーチンと %ZSTOP ルーチンによる開始動作と停止動作のカスタマイズ](#)" を参照してください。

LicenseID

インスタンスがライセンス・サーバにキーを要求できるようにします。

構文

```
[Startup]      LicenseID=n
```

n は、ターゲット **.key** ファイル内のライセンス・キーの名前です。既定では、キーは指定されていません。

説明

インスタンスの開始時にローカルの **iris.key** ファイルが検出されない場合、InterSystems IRIS® データ・プラットフォームは、**LicenseID** を使用してライセンス・サーバにライセンス・キーを要求します。ライセンス・サーバにロードされた各ライセンス・キーには固有の **LicenseID** が付きます。

ライセンス・キーの詳細は、“[InterSystems IRIS ライセンスの管理](#)” を参照してください。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[LicenseID]** の行の **[編集]** を選択します。**LicenseID** を入力します。

管理ポータルを使用する代わりに、**Config.Startup** クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、**LicenseID** を変更できます。

MaxConsoleLogSize

messages.log ファイルの最大サイズを設定します。

構文

```
[Startup]      MaxConsoleLogSize=n
```

n は、1 から 500 (MB) の範囲の整数です。既定値は 5 です。

説明

MaxConsoleLogSize は、InterSystems IRIS® Data Platform コンソール・ファイルの最大サイズ (MB) です。コンソール・ファイル **messages.log** は、既定の `install-dir¥mgr` ディレクトリ、または **console** パラメータで指定されたディレクトリにあります。

MaxConsoleLogSize の現在の設定よりも小さい値を入力した場合、またはコンソール・ファイルが増加してサイズ制限に達した場合、現在の **messages.log** ファイルの名前は **messages.old_Date** に変更されます。続いて、空の **messages.log** ファイルが作成され、新規のエントリは、新しく作成されたファイルに追加されます。

メッセージ・ログは、管理ポータルの **[メッセージログ]** ページ (**[システム処理]** > **[システムログ]** > **[メッセージログ]**) で参照できます。**messages.log** ファイルの場所を構成するには、**ConsoleFile** パラメータを参照してください。

このパラメータの変更

管理ポータルの **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[MaxConsoleLogSize]** の行の **[編集]** を選択します。MB 単位の数値を入力します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("**アクティブな CPF の編集**" を参照)、MaxConsoleLogSize を変更できます。

MaxIRISTempSizeAtStart

再起動時に IRISTEMP データベースの最大サイズを設定します。

構文

```
[Startup]      MaxIRISTempSizeAtStart=n
```

n は、0 から 1,000,000 (MB) の範囲の整数です。既定値は 0 です。

説明

MaxIRISTempSizeAtStart は、インスタンス起動時の **IRISTEMP** データベースの最大物理サイズ (MB 単位) です。インスタンスの再起動時に、InterSystems IRIS® データ・プラットフォームは、**IRISTEMP** データベースを MaxIRISTempSizeAtStart で指定されたサイズに切り捨てます。このパラメータが 0 に設定されている場合、インスタンスの再起動時に **IRISTEMP** データベースは切り捨てられません。

注釈 InterSystems IRIS はインスタンスの起動時に、**IRISTEMP** データベースで最大 240 MB を初期化します。これは、データベースの **IRIS.DAT** ファイルの物理サイズより大幅に小さい可能性があります。

IRISTEMP の [空き容量情報](#) を調べて **IRISTEMP** の論理サイズを確認し、**IRISTEMP** が増大しすぎている場合は、管理ポータルから手動で [データベースを切り捨てる](#) ことができます。

このパラメータの変更

管理ポータルの **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[MaxIRISTempSizeAtStart]** の行の **[編集]** を選択します。MB 単位の数値を入力します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、MaxIRISTempSizeAtStart を変更できます。

PasswordHash

パスワードの暗号化ハッシュとそのソルトを使用して、事前定義のユーザ・アカウントの既定のパスワードを設定します。

構文

```
[Startup] PasswordHash=a,b[,c,d]
```

a はハッシュ化されたパスワードです。b はそのソルトで、必要に応じて、c および d で、パスワードのハッシュ化に使用される作業係数およびアルゴリズムを指定できます。

説明

PasswordHash の主な用途は、自動導入で[事前定義のユーザ・アカウント](#)に既定のパスワードを設定することです。PasswordHash は、ハッシュ化されたパスワードとそのソルトを指定し、必要に応じて、そのパスワードのハッシュ化に使用される作業係数 (既定値 10000) とハッシュ化アルゴリズム (既定値 SHA512) も指定します。これを CPF マージ・ファイルで使用すると、インスタンスの起動時に、1 つ以上のロールが割り当てられている有効なユーザ・アカウント (つまり、ロールが割り当てられない **CSPSystem** を除くすべての事前定義のアカウント) ごとに、格納されたパスワード・ハッシュが PasswordHash の値に設定されます。PasswordHash が空の場合、これは CPF マージ操作で無視されます。

重要 PasswordHash プロパティを使用できるのは、特定の InterSystems IRIS インスタンスに対して一度のみで、なおかつどの事前定義アカウントに対しても既定のパスワードが変更されていない場合のみです。導入後に既定のパスワードを変更しないままにすることが可能であることは重大なセキュリティ・リスクであるため、構成マージ操作では PasswordHash 設定を使用して導入時に (それ以降ではなく) 既定のパスワードを変更する必要があります (個々のユーザのパスワードを変更する方法は、[「既存のユーザ・アカウントの編集」](#) を参照してください)。

導入済みインスタンスのパスワードを変更する目的には、PasswordHash は推奨されません。

例

コンテナの導入時におけるパスワードのハッシュ化の詳細と PasswordHash の使用例は、[「認証とパスワード」](#) を参照してください。

関連項目

- ・ [認証とパスワード](#)
- ・ [インスタンス認証](#)
- ・ [既存のユーザ・アカウントの編集](#)

ProcessHalt

フォアグラウンド・プロセスのシャットダウン中にカスタム・ルーチンを実行できるようにします。

構文

```
[Startup]      ProcessHalt=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

ProcessHalt が有効な場合 (n = 1)、InterSystems IRIS® Data Platform は、フォアグラウンド・プロセスからのログアウト時 (ユーザがターミナルを閉じたときなど) に LOGIN^%ZSTOP ルーチン・エントリを実行します。このパラメータが有効ではない場合、ルーチンは実行されません。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[ProcessHalt]** の行の **[編集]** を選択します。**[ProcessHalt]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、ProcessHalt を変更できます。

関連項目

[ProcessStart](#) および [ProcessHalt](#) は、フォアグラウンド・プロセスに対応します。これらは、ターミナル・セッションを介して起動されるか、フォアグラウンドで実行するために設定されるプロセスです。

[JobStart](#) および [JobHalt](#) は、バックグラウンド・プロセスに対応します。これには、JOB コマンドを介して起動されるプロセスと、ODBC、任意の言語バインディングなどのバックグラウンド・サーバ・プロセスとが含まれます。

[SystemStart](#) および [SystemHalt](#) は、InterSystems IRIS インスタンスの開始またはシャットダウンに対応します。

[CallinStart](#) および [CallinHalt](#) は、CALLIN を実行する外部プログラムに対応します。

これらすべてのパラメータの詳細は、"[%ZSTART ルーチンと %ZSTOP ルーチンによる開始動作と停止動作のカスタマイズ](#)" を参照してください。

ProcessStart

フォアグラウンド・プロセスの開始中にカスタム・ルーチンを実行できるようにします。

構文

```
[Startup]      ProcessStart=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

ProcessStart が有効な場合 (n = 1)、InterSystems IRIS® Data Platform は、フォアグラウンド・プロセスへのログイン時 (ユーザがターミナルにログインしたときなど) に LOGIN`%ZSTART ルーチン・エントリを実行します。このパラメータが有効ではない場合、ルーチンは実行されません。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[ProcessStart]** の行の **[編集]** を選択します。**[ProcessStart]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、ProcessStart を変更できます。

関連項目

ProcessStart および [ProcessHalt](#) は、フォアグラウンド・プロセスに対応します。これらは、ターミナル・セッションを介して起動されるか、フォアグラウンドで実行するために設定されるプロセスです。

[JobStart](#) および [JobHalt](#) は、バックグラウンド・プロセスに対応します。これには、JOB コマンドを介して起動されるプロセスと、ODBC、任意の言語バインディングなどのバックグラウンド・サーバ・プロセスとが含まれます。

[SystemStart](#) および [SystemHalt](#) は、InterSystems IRIS インスタンスの開始またはシャットダウンに対応します。

[CallinStart](#) および [CallinHalt](#) は、CALLIN を実行する外部プログラムに対応します。

これらすべてのパラメータの詳細は、"[`%ZSTART ルーチンと `%ZSTOP ルーチンによる開始動作と停止動作のカスタマイズ](#)" を参照してください。

ShutdownTimeout

強制的にシャットダウンされるまで InterSystems IRIS® Data Platform が待機する秒数を設定します。

構文

```
[Startup] ShutdownTimeout=n
```

n は、120 から 100,000 の範囲の整数です。既定値は 300 (5 分) です。

説明

ShutdownTimeout は、シャットダウンの正常完了を InterSystems IRIS が待機する時間 (秒) です。この時間に達するとタイムアウトになり、強制的にシャットダウンが行われます。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[ShutdownTimeout]** の行の **[編集]** を選択します。秒数を入力します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、ShutdownTimeout を変更できます。

SystemHalt

インスタンスのシャットダウン時にカスタム・ルーチンを実行できるようにします。

構文

```
[Startup]    SystemHalt=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

SystemHalt が有効な場合 (n = 1)、InterSystems IRIS® データ・プラットフォームは、インスタンスのシャットダウン時に SYSTEM^%ZSTOP ルーチン・エントリを実行します。このパラメータが有効ではない場合、ルーチンは実行されません。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[SystemHalt]** の行の **[編集]** を選択します。**[SystemHalt]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、SystemHalt を変更できます。

関連項目

[ProcessStart](#) および [ProcessHalt](#) は、フォアグラウンド・プロセスに対応します。これらは、ターミナル・セッションを介して起動されるか、フォアグラウンドで実行するために設定されるプロセスです。

[JobStart](#) および [JobHalt](#) は、バックグラウンド・プロセスに対応します。これには、JOB コマンドを介して起動されるプロセスと、ODBC、任意の言語バインディングなどのバックグラウンド・サーバ・プロセスとが含まれます。

[SystemStart](#) および [SystemHalt](#) は、InterSystems IRIS インスタンスの開始またはシャットダウンに対応します。

[CallinStart](#) および [CallinHalt](#) は、CALLIN を実行する外部プログラムに対応します。

これらすべてのパラメータの詳細は、"[%ZSTART ルーチンと %ZSTOP ルーチンによる開始動作と停止動作のカスタマイズ](#)" を参照してください。

SystemMode

管理ポータルへのヘッダに表示されるラベルを指定します。

構文

```
[Startup]      SystemMode=n
```

n は英数字文字列です。最大長は 32 文字です。既定値は空文字列です。

説明

SystemMode は、[管理ポータルへのヘッダ](#)に表示されるラベルを指定します。このラベルを使用すると、InterSystems IRIS® Data Platform のインスタンスを簡単に特定できます。

任意のラベルを入力できます。ただし、一部の SystemMode 値は、管理ポータルで特別な方法で扱われます。これらの値は以下のとおりです。

- ・ LIVE – [本稼働システム] に変換され、赤いテキストと赤い境界線が付きます。
- ・ TEST – [テストシステム] に変換されます。
- ・ DEVELOPMENT – [開発システム] に変換されます。
- ・ FAILOVER – [フェイルオーバー・システム] に変換されます。

このパラメータの変更

管理ポータルの [開始] ページ ([システム管理] > [構成] > [追加設定] > [開始]) で、[SystemMode] の行の [編集] を選択します。[SystemMode] テキスト・ボックスに目的のラベルを入力して、[保存] をクリックします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、SystemMode を変更できます。

SystemStart

インスタンスの開始時にカスタム・ルーチンを実行できるようにします。

構文

```
[Startup]    SystemStart=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

SystemStart が有効な場合 (n = 1)、InterSystems IRIS® データ・プラットフォームは、インスタンスの開始時に SYS-TEM^%ZSTART ルーチン・エントリを実行します。このパラメータが有効ではない場合、ルーチンは実行されません。

このパラメータの変更

管理ポータル の [開始] ページ ([システム管理] > [構成] > [追加設定] > [開始]) で、[SystemStart] の行の [編集] を選択します。[SystemStart] を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、SystemStart を変更できます。

関連項目

[ProcessStart](#) および [ProcessHalt](#) は、フォアグラウンド・プロセスに対応します。これらは、ターミナル・セッションを介して起動されるか、フォアグラウンドで実行するために設定されるプロセスです。

[JobStart](#) および [JobHalt](#) は、バックグラウンド・プロセスに対応します。これには、JOB コマンドを介して起動されるプロセスと、ODBC、任意の言語バインディングなどのバックグラウンド・サーバ・プロセスとが含まれます。

SystemStart および [SystemHalt](#) は、InterSystems IRIS インスタンスの開始またはシャットダウンに対応します。

[CallinStart](#) および [CallinHalt](#) は、CALLIN を実行する外部プログラムに対応します。

これらすべてのパラメータの詳細は、"[%ZSTART ルーチンと %ZSTOP ルーチンによる開始動作と停止動作のカスタマイズ](#)" を参照してください。

TempDirectory

一時ファイルのサブディレクトリを指定します。

構文

```
[Startup]      TempDirectory=n
```

n は、絶対ディレクトリ・パス名または相対ディレクトリ・パス名です。TempDirectory の値が有効なパス名の場合、最大長はありません。既定値は Temp で、これは <install-dir>%mgr%Temp に対応します。

説明

TempDirectory は、InterSystems IRIS® Data Platform が一時ファイルを保存するサブディレクトリの名前です。TempDirectory に新しい値を設定すると、この名前のサブディレクトリが作成され、InterSystems IRIS の新しい一時ディレクトリになります。

パスの指定は、フル・パスでも相対パスでもかまいません。フル・パスを指定した場合、指定したディレクトリが使用されます。相対パスを指定した場合、そのディレクトリが <install-dir>%mgr% サブディレクトリの下位に作成されます。

例

Windows で c:%InterSystems%iris%mgr%Temp% を作成するには、以下のように指定します。

```
TempDirectory=Temp
```

Windows で c:%TempFiles% を作成するには、以下のように指定します。

```
TempDirectory=c:\TempFiles
```

このパラメータの変更

管理ポータルの [開始] ページ ([システム管理] > [構成] > [追加設定] > [開始]) で、[TempDirectory] の行の [編集] を選択します。サブディレクトリ名を入力します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、TempDirectory を変更できます。

TerminalPrompt

ターミナル・プロンプトの書式を定義します。

構文

```
[Startup]      TerminalPrompt=n
```

n は、ターミナル・プロンプトを設定するコンマ区切り値 (0-8) の文字列です。既定値は 8,2 です。

説明

TerminalPrompt は、インスタンスのターミナル・プロンプトに表示される情報を構成します。文字列内の値の順序により、プロンプトに表示される情報の順序が決定されます。例えば、文字列 2,1 はターミナル・プロンプト %SYS:HostName> を生成します。

値は以下のようにコード化されています。

- ・ 0 – 単純なプロンプト。プロンプトで “>” のみを使用するには、他の値を指定せずに 0 を指定します。
- ・ 1 – ホスト名。現在のシステム名ともいいます。コンピュータに割り当てられた名前です。例えば、LABLAPTOP> です。これは、すべてのターミナルのプロセスで同じです。
- ・ 2 – ネームスペース名。例えば、%SYS> です。現在のネームスペース名は \$NAMESPACE 特殊変数に格納されます。このネームスペース名は、明示的でも暗黙的でもかまいません。
- ・ 3 – 構成名。インスタンスの名前。例えば、IRIS2> です。これは、すべてのターミナルのプロセスで同じです。
- ・ 4 – 現在の時刻。24 時間形式 (整数秒単位) のローカル時刻で示されます。例えば、15:59:36> です。これは、プロンプトが返されたときの静的な時刻値になります。この値は、プロンプトごとに変わります。
- ・ 5 – ターミナルのプロセス ID。例えば、2336> です。これは、ターミナルのプロセスごとに異なります。この値は、\$JOB 特殊変数から返すこともできます。
- ・ 6 – ユーザ名。例えば、fred> です。これは、すべてのターミナルのプロセスで同じです。
- ・ 7 – 最後のコマンドを実行している経過時間 (秒.ミリ秒単位)。例えば、.000495> です。先頭および末尾のゼロは抑制されます。この値は、プロンプトごとに変わります。
- ・ 8 – トランザクション・レベル。例えば、TL1> です。

TerminalPrompt を空白のままにすることはできません。入力文字列が無効な場合、ターミナルは TerminalPrompt の既定値 8,2 を使用します。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[TerminalPrompt]** の行の **[編集]** を選択します。コンマで区切られた値の文字列、または 0 を入力します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、TerminalPrompt を変更できます。

WebServer

プライベート Web サーバを起動できるようにします (存在する場合)。

構文

重要 2023.2 より前のバージョンの InterSystems IRIS® には、管理ポータルなどの組み込みの Web アプリケーションを処理するプライベート Web サーバが含まれていました。2023.2 以降、InterSystems IRIS の新規インストールにはプライベート Web サーバは含まれなくなりました。詳細は、“Web ゲートウェイ・ガイド”の[こちらのセクション](#)を参照してください。

```
[Startup]      WebServer=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は、新規インストールの場合は 0、アップグレードの場合は 1 です。

説明

WebServer が有効な場合 (n = 1)、InterSystems IRIS は起動時に Apache プライベート Web サーバの起動を試みます。

このパラメータの変更

管理ポータルの **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[WebServer]** の行の **[編集]** を選択します。**[WebServer]** を選択して、プライベート Web サーバを有効にします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)”を参照)、WebServer を変更できます。

注釈 この設定を有効にするには、再起動が必要です。

関連項目

- [WebServerName](#)
- [WebServerPort](#)
- [WebServerProtocol](#)
- [WebServerURLPrefix](#)

WebServerName

Web サーバの IP アドレスまたは DNS 名を特定します。

構文

```
[Startup]    WebServerName=nnn.nnn.nn.nn (ip address) or www.DNSname.com
```

nnn.nnn.nn.nn は有効な IP アドレスです。www.DNSname.com は有効な DNS 名です。1 つの値だけを指定します。

説明

WebServerName は、InterSystems IRIS® データ・プラットフォーム・ツールで使用するよう構成される Web サーバの DNS 名または IP アドレスを特定します。このパラメータを設定する必要があるのは、InterSystems スタジオを有効にする必要がある場合のみです。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[WebServerName]** の行の **[編集]** を選択します。DNS 名または IP アドレスを入力します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、WebServerName を変更できます。

関連項目

- [WebServer](#)
- [WebServerPort](#)
- [WebServerProtocol](#)
- [WebServerURLPrefix](#)

WebServerPort

Web サーバのポートを特定します。

構文

```
[Startup]      WebServerPort=nnnnn
```

n は有効なポート番号です。既定値は 80 です。

説明

WebServerPort は、InterSystems IRIS® データ・プラットフォーム・ツールで使用するよう構成される Web サーバのポート番号を特定します。このパラメータを設定する必要があるのは、InterSystems スタジオを有効にする必要がある場合のみです。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[WebServerPort]** の行の **[編集]** を選択します。ポート番号を入力します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、WebServerPort を変更できます。

関連項目

- [WebServer](#)
- [WebServerName](#)
- [WebServerProtocol](#)
- [WebServerURLPrefix](#)

WebServerProtocol

InterSystems IRIS がカスタム・ポート経由で HTTPS 接続を受信できるようにします。

構文

```
[Startup]      WebServerProtocol=protocol
```

protocol は、http (既定値) または https のいずれかです。

説明

WebServerPort を 443 (HTTPS の標準ポート) に設定すると、スタジオは自動的に HTTPS を使用します。その他の値に設定した場合、スタジオが HTTPS 経由で通信できるようにするには WebServerProtocol を https に設定する必要があります。このパラメータを設定する必要があるのは、InterSystems スタジオを有効にする必要がある場合のみです。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加の設定]** > **[開始]**) で、**[WebServerProtocol]** の行の **[編集]** を選択します。https または http と入力します。

管理ポータルを使用する代わりに、Config.Startup クラスで (クラス・リファレンスを参照)、またはテキスト・エディタで CPF を編集することによって (このドキュメントの “構成パラメータ・ファイルの概要” の章にある “[アクティブな CPF の編集](#)” のセクションを参照)、WebServerProtocol を変更できます。

関連項目

- [WebServer](#)
- [WebServerName](#)
- [WebServerPort](#)
- [WebServerURLPrefix](#)

WebServerSSLConfiguration

スタジオに実装されている CSP テンプレートを使用して作成した %Net.HttpRequest オブジェクトの既定の TLS 構成を指定します。

構文

```
[Startup]    WebServerSSLConfiguration=configuration
```

configuration は、TLS 構成の場所です。

説明

重要 このパラメータは互換性の目的でのみ保持されています。新しいアプリケーションの構築では使用しないでください。

非推奨。HTTPS 接続のみを受け入れるアクティブなプライベート Web サーバが設定されているインスタンスで、スタジオを使用して CSP テンプレートから Web アプリケーションを作成する場合は、指定する必要があります。

このパラメータの変更

管理ポータル の [開始] ページ ([システム管理] > [構成] > [追加の設定] > [開始]) で、[WebServerProtocol] の行の [編集] を選択します。TLS 構成の場所を入力します。

管理ポータルを使用する代わりに、Config.Startup クラスで (クラス・リファレンスを参照)、またはテキスト・エディタで CPF を編集することによって (このドキュメントの “構成パラメータ・ファイルの概要” の章にある “[アクティブな CPF の編集](#)” のセクションを参照)、WebServerSSLConfiguration を変更できます。

関連項目

- [WebServer](#)
- [WebServerName](#)
- [WebServerPort](#)
- [WebServerProtocol](#)
- [WebServerURLPrefix](#)

WebServerURLPrefix

Web サーバが InterSystems IRIS® データ・プラットフォームへのアクセスに使用するインスタンス接頭語を特定します。

構文

```
[Startup]      WebServerURLPrefix=n
```

n は、URL で使用する英数字文字列です。ガイドラインとして、WebServerURLPrefix は 80 文字未満にすることをお勧めします。既定値は、すべて小文字のインスタンス名です。

説明

WebServerURLPrefix は、URL を構成する際にスタジオによって使用されます。このパラメータを設定する必要があるのは、InterSystems スタジオを有効にする必要がある場合のみです。[\[CSP サーバインスタンス\]](#) 設定に一致している必要があります。

このパラメータの変更

管理ポータルでの [\[開始\]](#) ページ ([\[システム管理\]](#) > [\[構成\]](#) > [\[追加設定\]](#) > [\[開始\]](#)) で、[\[WebServerURLPrefix\]](#) の行の [\[編集\]](#) を選択します。InterSystems IRIS インスタンス名を入力します。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、WebServerURLPrefix を変更できます。

関連項目

- [WebServer](#)
- [WebServerName](#)
- [WebServerPort](#)
- [WebServerProtocol](#)

ZSTU

ユーザ定義の起動を実行できるようにします。

構文

```
[Startup]      ZSTU=n
```

n は、1 (真) または 0 (偽) のいずれかです。既定値は 1 です。

説明

重要 このパラメータは互換性のために保持されていますが、`^ZSTU` ルーチンの使用は推奨されません。代わりに、`%ZSTART` ルーチンを使用してください。

ZSTU が有効な場合 (n = 1)、InterSystems IRIS® Data Platform では、`^ZSTU` ルーチンからユーザ定義の起動が実行されます。

このパラメータの変更

管理ポータル の **[開始]** ページ (**[システム管理]** > **[構成]** > **[追加設定]** > **[開始]**) で、**[ZSTU]** の行の **[編集]** を選択します。**[ZSTU]** を選択して、この設定を有効にします。

管理ポータルを使用する代わりに、Config.Startup クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、ZSTU を変更できます。

[Telnet]

このトピックでは、CPF の [Telnet] セクションのパラメータについて説明します。これらの設定は、インターシステムズが Telnet サーバを提供している Windows 構成にのみ適用されます。Telnet がオペレーティング・システム・ベンダから提供されている UNIX®、または Linux の構成には適用されません。

これらの設定は、管理ポータルの [Telnet 設定] ページにもあります ([システム管理] > [構成] > [デバイス設定] > [Telnet 設定])。

DNSLookup

Telnet クライアント・アドレスを DNS で検索できるようにします。Windows システムでのみ有効です。

構文

```
[Telnet]      DNSLookup=n
```

n は、ON または OFF のいずれかの文字列です。既定値は ON です。

説明

DNSLookup では、接続を処理するために作成された InterSystems IRIS® Data Platform プロセスにアドレスを渡す前に、Telnet デーモンでのクライアント・アドレスの DNS 検索を有効または無効にします。これにより、InterSystems IRIS プロセスの \$IO と \$ZIO によって返されるクライアント・アドレスの形式が決まります。

DNSLookup が有効な場合、クライアント・アドレスの DNS 検索が実行され、クライアント名が InterSystems IRIS に渡されます。DNSLookup が有効ではない場合、DNS 検索が実行されないため、クライアント・アドレスは、ドットで区切った 10 進表記 (IPv4 による接続の場合)、またはコロンで区切った 16 進表記 (IPv6 による接続の場合) の形式で得られます。DNS サーバで検索ができない場合は、このパラメータを無効にする必要があります。DNS サーバが利用できない場合は、ログインに時間がかかるためです。

InterSystems IRIS Telnet 設定は、インターシステムズが Telnet サーバを提供している Windows 構成にのみ適用されます。このパラメータは、UNIX® システムでは無視されます。

このパラメータの変更

管理ポータル の [Telnet 設定] ページ ([システム管理] > [構成] > [デバイス設定] > [Telnet 設定]) で、[DNS 参照] フィールドで [オン] または [オフ] を選択します。

管理ポータルを使用する代わりに、Config.Telnet クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、DNSLookup を変更できます。

この設定を編集した場合、変更内容を適用するには、InterSystems IRIS を再起動する必要があります。

Port

Telnet ポート番号を設定します。Windows システムでのみ有効です。

構文

```
[Telnet]      Port=n
```

n は、有効な TCP/IP ポート番号です。既定値は 23 です。

説明

Port は、Telnet 接続用の TCP/IP ポート番号です。同じホストで複数の InterSystems IRIS® Data Platform 構成を実行する場合は、各構成でそれぞれ異なる Telnet ポート番号を指定する必要があります。クライアントは、Telnet をクライアント・システムで起動する場合、ポート番号を指定することによって、既定値以外のポート番号を使用した構成を実行できます。Telnet は、SSL を使用するかどうかにかかわらず任意のポートで構成でき、ポート 992 を使用する必要はありません。

InterSystems IRIS Telnet 設定は、インターシステムズが Telnet サーバを提供している Windows 構成にのみ適用されます。このパラメータは、UNIX® システムでは無視されます。

このパラメータの変更

管理ポータル の [Telnet 設定] ページ ([システム管理] > [構成] > [デバイス設定] > [Telnet 設定]) で、[Telnet ポート番号] フィールドに TCP/IP ポート番号を入力します。

管理ポータルを使用する代わりに、Config.Telnet クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって ("[アクティブな CPF の編集](#)" を参照)、Port を変更できます。

[WorkQueues]

このトピックでは、CPF の [WorkQueues] セクションの作業キュー・マネージャ・カテゴリについて説明します。既存のカテゴリと同じ構文を使用して新しいカテゴリを作成できます。作業キュー・マネージャの詳細は、“[作業キュー・マネージャの使用](#)”を参照してください。

Default

“既定”の作業キュー・マネージャ・カテゴリを定義します。

構文

```
[WorkQueues]      Default=MaxActiveWorkers,DefaultWorkers,MaxWorkers
```

Default は、この作業キュー・マネージャ・カテゴリの名前です。MaxActiveWorkers、DefaultWorkers、および MaxWorkers は、3 つのコンマ区切りの整数です。これらは、空白のままにするか、0 を指定すると、既定値を使用できます。

MaxActiveWorkers と MaxWorkers に既定値を使用すると、作業キュー・マネージャ・カテゴリの MaxActiveWorkers の数字がパラメータに対する唯一のハード・リミットとなります。MaxActiveWorkers に値 -1 を指定することによって、作業キュー・マネージャ・カテゴリでアクティブ・ワーカが無制限に許容されます。

説明

構成パラメータ・ファイル (CPF) の [WorkQueues] セクションには、各作業キュー・マネージャ・カテゴリのエントリが含まれます。カテゴリには最大 3 つのコンマ区切りプロパティが含まれます。どの引数も必須ではありません。空白のままにすると、既定値として解決されます。詳細は、“[カテゴリの管理](#)”を参照してください。

このパラメータの変更

管理ポータルでの [Work Queue Manager Categories] ページ ([システム管理] > [構成] > [システム構成] > [WQM Categories]) で、[デフォルト] をクリックして、このカテゴリを編集します。代わりに、[Create Category] をクリックして、新しい WQM カテゴリを作成することもできます。

管理ポータルを使用する代わりに、Config.WorkQueues クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)”を参照)、Default カテゴリを変更できます。

関連項目

- ・ [作業キュー・マネージャの使用](#)
- ・ [作業キュー・マネージャ・カテゴリの作成](#)

SQL

“SQL” の作業キュー・マネージャ・カテゴリを定義します。

構文

```
[WorkQueues]      SQL=MaxActiveWorkers,DefaultWorkers,MaxWorkers
```

SQL は、この作業キュー・マネージャ・カテゴリの名前です。MaxActiveWorkers、DefaultWorkers、および MaxWorkers は、3 つのコンマ区切りの整数です。これらは、空白のままにするか、0 を指定すると、既定値を使用できます。MaxActiveWorkers と MaxWorkers に既定値を使用すると、作業キュー・マネージャ・カテゴリの MaxActiveWorkers の数字がパラメータに対する唯一のハード・リミットとなります。MaxActiveWorkers に値 -1 を指定することによって、作業キュー・マネージャ・カテゴリでアクティブ・ワーカが無制限に許容されます。

説明

構成パラメータ・ファイル (CPF) の [WorkQueues] セクションには、各作業キュー・マネージャ・カテゴリのエントリが含まれます。カテゴリには最大 3 つのコンマ区切りプロパティが含まれます。どの引数も必須ではありません。空白のままにすると、既定値として解決されます。詳細は、“[カテゴリの管理](#)” を参照してください。

このパラメータの変更

管理ポータル の [Work Queue Manager Categories] ページ ([システム管理] > [構成] > [システム構成] > [WQM Categories]) で、[SQL] をクリックして、このカテゴリを編集します。代わりに、[Create Category] をクリックして、新しい WQM カテゴリを作成することもできます。

管理ポータルを使用する代わりに、Config.WorkQueues クラス (クラス・リファレンスを参照) で、またはテキスト・エディタで CPF を編集することによって (“[アクティブな CPF の編集](#)” を参照)、SQL カテゴリを変更できます。

関連項目

- ・ [作業キュー・マネージャの使用](#)
- ・ [作業キュー・マネージャ・カテゴリの構成](#)

